



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Desarrollo de sistema autónomo basado en la plataforma
LEGO EV3 para la resolución de juegos matemáticos.

Autor/es

ANA RÍO RODRÍGUEZ

Director/es

JAVIER RICO AZAGRA

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2018-19



Desarrollo de sistema autónomo basado en la plataforma LEGO EV3 para la resolución de juegos matemáticos., de ANA RÍO RODRÍGUEZ

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor, 2019

© Universidad de La Rioja, 2019

publicaciones.unirioja.es

E-mail: publicaciones@unirioja.es



**UNIVERSIDAD
DE LA RIOJA**

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE GRADO

**TITULACIÓN: Grado en
Ingeniería Electrónica Industrial y Automática**

CURSO: 2018/2019

CONVOCATORIA: JULIO

TÍTULO:

**Desarrollo de sistema autónomo basado en la plataforma
Lego EV3 para la resolución de juegos matemáticos**

AUTOR: Ana Río Rodríguez

DIRECTOR/ES: Javier Rico Azagra

DEPARTAMENTO: Ingeniería Eléctrica



RESUMEN

En este proyecto se desarrolla tanto el hardware como el software para la construcción de un robot capaz de resolver juegos matemáticos de tipo *sudoku*. Una vez alcanzada la solución el sistema es capaz de escribir sobre un papel la solución de forma autónoma. En el presente documento, además se incluye un estudio previo de posibles soluciones, el presupuesto, los planos necesarios para acometer la solución alcanzada y el pliego de condiciones correspondientes al proyecto.

En cuanto al hardware, dicho robot encargado de transcribir las soluciones en papel es construido empleando componentes del kit LEGO Mindstorms Education. En concreto se han empleado sensores táctiles y tres motores rotativos, además de un brick de control y una serie de piezas que facilitan el montaje.

Por otro lado, el software, es desarrollado íntegramente en Matlab y su herramienta Simulink. Los datos de partida del problema a resolver (*sudoku*), son introducidos por el usuario en el código y posteriormente el robot se encarga de la resolución del mismo. Una vez alcanzada una solución, una estructura de control jerárquica formada por múltiples lazos de control anidados permite que el robot escriba los números necesarios en las casillas correspondientes. Para ello se emplea un rotulador y un mecanismo de cinco eslabones con dos grados de libertad.

La comunicación entre el robot y el PC que contiene en el que se ejecutan los algoritmos necesarios se realiza a través de red WiFi. Esta comunicación se ejecuta en tiempo real, es decir, en todo momento el PC sabe qué está ejecutando el robot y viceversa.

ABSTRACT

This project develops both hardware and software for the construction of a robot capable of solving the mathematical games known as *sudoku*. Once the solution is reached, the system is able to write the solution autonomously on a paper. This document includes a preliminary study of possible solutions, the budget, the necessary drawings to undertake the solution reached and the specifications of the project.

In terms of hardware, the robot, responsible for transcribing paper solutions is built using components of the Lego Mindstorms Education kit. In particular, touch sensors and three rotary motors have been used, in addition to a control brick and a series of parts that make the assembly possible.

On the other hand, the software is developed entirely in Matlab and its Simulink tool. The starting data of the problem to be solved (*sudoku*), are introduced by the user in the code and then the robot is responsible for the resolution of the same. Once a solution is reached, a hierarchical control structure formed by multiple nested control loops allows the robot to write the necessary numbers in the corresponding boxes. For this, a marker and a five-link mechanism with two degrees of freedom are used.

The communication between the robot and the PC that contains the necessary algorithms, is carried out through the WiFi network. This communication is executed in real time, that means, the PC knows what the robot is running at all times and vice versa.

AGRADECIMIENTOS

Este trabajo fin de grado y el grado en general ha sido un camino duro, con obstáculos y altibajos. Pero la satisfacción de superarlos era el combustible que me hacía seguir adelante.

No he estado sola en este camino, y ha habido muchas personas ayudándome a superar esos obstáculos. A esas personas van dirigidas estos agradecimientos.

En primer lugar, a mi familia, ellos son los que más me han sufrido cuando estaba agobiada y también los que han tenido que aguantar explicaciones eternas cada vez que conseguía dar un pasito en este proyecto. Porque me hacía ilusión contárselo, a pesar de que en el fondo no lo entendían, y tampoco se molestaban en ello, pero ponían un gesto alegre al ver mi cara de emoción.

En segundo lugar, agradecer a Javier Rico Azagra, tutor de este proyecto, sus consejos y sabiduría. Sin él este proyecto no hubiese sido posible.

Y por último a toda esa gente que directa o indirectamente ha ayudado a que yo llegue a este punto, gracias.





**UNIVERSIDAD
DE LA RIOJA**

Documento 1:

ÍNDICE GENERAL

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

1. ÍNDICE GENERAL

| | |
|---|---------------|
| RESUMEN | 2 |
| ABSTRACT | 3 |
| AGRADECIMIENTOS | 4 |
| ÍNDICE GENERAL | 6 |
| 1. ÍNDICE GENERAL | 7 |
| 2. ÍNDICE DE ILUSTRACIONES..... | 10 |
| 3. ÍNDICE DE TABLAS | 12 |
| MEMORIA | 14 |
| ÍNDICE DE LA MEMORIA | 15 |
| 1. INTRODUCCIÓN | 17 |
| 2. OBJETO..... | 18 |
| 3. ALCANCE..... | 20 |
| 4. ANTECEDENTES..... | 21 |
| 5. ABREVIATURAS Y DEFINICIONES..... | 23 |
| 6. NORMAS Y REFERENCIAS | 24 |
| 6.1. Disposiciones legales y normativa aplicada..... | 24 |
| 6.2. Programas utilizados..... | 24 |
| 7. BIBLIOGRAFÍA | 25 |
| 7.1. Enlaces web | 25 |
| 7.2. Libros bibliográficos..... | 25 |
| 8. ANÁLISIS DE POSIBLES SOLUCIONES | 26 |
| 8.1. Punto de vista Hardware | 26 |
| 8.2. Punto de vista Software | 26 |
| 9. SOLUCIÓN ELEGIDA | 28 |
| 10. RESOLUCIÓN DEL SUDOKU | 29 |
| 10.1. Introducción histórica | 29 |
| 10.2. Reglas para la resolución | 29 |
| 10.3. Algoritmo utilizado para la resolución | 30 |
| 10.4. Diagrama de flujo de la resolución | 32 |
| 10.5. Ejemplo de resolución | 33 |
| 11. DIMENSIONES FÍSICAS..... | 35 |
| 11.1. Dimensiones de los brazos del robot | 35 |
| 11.2. Dimensiones del Sudoku | 36 |
| 11.3. Dimensiones de los números | 36 |
| 11.4. Coordenadas del origen de las casillas..... | 37 |
| 12. HARDWARE..... | 38 |
| 12.1. Componentes del kit LEGO Mindstorms Education EV3 utilizados..... | 38 |

| | |
|---|-----------|
| 12.2. Diseño e impresión 3D | 40 |
| 12.3. Relaciones de transmisión | 41 |
| 13. SOFTWARE | 44 |
| 13.1. Esquema general | 44 |
| 13.2. Lazos de control de los motores | 44 |
| 13.3. Obtención de las plantas y controladores de los motores | 48 |
| 13.4. Cálculo de ángulos a partir del punto | 50 |
| 13.5. Error de aproximación al punto | 56 |
| 13.6. Obtención de coordenadas | 57 |
| 13.7. Obtención del número y casilla a partir del sudoku solución | 61 |
| 14. COMUNICACIÓN | 63 |
| 14.1. Puertos utilizados | 63 |
| 14.2. Tipos de conexión | 63 |
| 14.3. Procedimiento de conexión inalámbrica Wi-Fi | 64 |
| 14.4. Parámetros en Simulink | 65 |
| 15. RESOLUCIÓN POR MEDIO DEL ROBOT | 66 |
| 15.1 Referencia en el folio | 66 |
| 15.2 Solución | 67 |
| 16. CONCLUSIONES | 68 |
| 17. VÍAS DE CONTINUACIÓN | 69 |
| ANEXOS | 71 |
| ÍNDICE DE ANEXOS | 72 |
| 1. ESQUEMAS SIMULINK | 73 |
| 1.1. Esquema principal | 73 |
| 1.2. Esquema “Obtención de ángulos y control motores” | 74 |
| 1.3. Esquema “Motor C” | 75 |
| 1.4. Esquema “Motor A” | 75 |
| 1.5. Esquema “Motor B” | 75 |
| 2. CÓDIGO MATLAB | 76 |
| 2.1. Resolución del sudoku | 76 |
| 2.2. Función “Datos a escribir” | 78 |
| 2.3. Función “Obtención de coordenadas” | 79 |
| 2.4. Función “Comparador” | 83 |
| 2.5. Función “Cálculo de ángulos a partir de coordenadas” | 83 |
| 2.6. Función “RESET” del motor C | 84 |
| 2.7. Función “RESET” del motor A | 84 |
| 3. DISEÑO DE LA TRAYECTORIA DE LOS NÚMEROS | 85 |
| 4. PLANTILLA PARA LA RESOLUCIÓN | 90 |
| 5. INSTRUCCIONES DE MONTAJE | 92 |

| | |
|---|------------|
| PLANOS..... | 120 |
| ÍNDICE DE PLANOS | 121 |
| PLIEGO DE CONDICIONES | 127 |
| ÍNDICE DEL PLIEGO DE CONDICIONES | 128 |
| 1. INTRODUCCIÓN AL PLIEGO DE CONDICIONES..... | 129 |
| 2. PROPIEDAD INTELECTUAL | 129 |
| 3. CONDICIONES GENERALES | 129 |
| 4. CONDICIONES ADMINISTRATIVAS | 130 |
| 5. NORMATIVA Y REGLAMENTACIÓN | 130 |
| 6. CONDICIONES FACULTATIVAS..... | 131 |
| 6.1. Dirección | 131 |
| 6.2. Libro de órdenes | 131 |
| 6.3. Modificaciones | 131 |
| 7. CONDICIONES DE MATERIALES Y EQUIPOS..... | 132 |
| 7.1. Bloque de programación Lego EV3 | 132 |
| 7.2. Equipo PC..... | 132 |
| 8. CONDICIONES ECONÓMICAS | 133 |
| 8.1. Errores en el proyecto | 133 |
| 8.2. Liquidación..... | 133 |
| 9. DISPOSICIÓN FINAL | 134 |
| MEDICIONES..... | 136 |
| ÍNDICE DE MEDICIONES | 137 |
| 1. CUADRO DE MEDICIONES | 138 |
| PRESUPUESTO..... | 140 |
| ÍNDICE DEL PRESUPUESTO | 141 |
| 1. CUADRO DE PRECIOS UNITARIOS..... | 142 |
| 2. PRESUPUESTOS PARCIALES | 143 |
| 2.1. Recursos humanos | 143 |
| 2.2. Licencias..... | 143 |
| 2.3. Equipos y materiales..... | 144 |
| 3. PRESUPUESTO TOTAL | 145 |

2. ÍNDICE DE ILUSTRACIONES

| | |
|--|----|
| Ilustración 1: Robot resuelve cubos de Rubik | 21 |
| Ilustración 2: Robot capaz de tocar la guitarra | 22 |
| Ilustración 3: Robot reloj digital..... | 22 |
| Ilustración 4: Modelo 3D del robot complete..... | 28 |
| Ilustración 5: Cuadrícula del Sudoku | 29 |
| Ilustración 6: Diagrama de flujo de la resolución del Sudoku..... | 32 |
| Ilustración 7: Sudoku ejemplo..... | 33 |
| Ilustración 8: Matriz SUDOKU..... | 33 |
| Ilustración 9: Matriz huecos | 34 |
| Ilustración 10: Solución en Matlab..... | 34 |
| Ilustración 11: Dimensiones de los brazos del robot | 35 |
| Ilustración 12: Cuadrícula dimensiones de los números. | 37 |
| Ilustración 13: Pieza sujeción rotulador 1. | 40 |
| Ilustración 14: Pieza sujeción rotulador 2. | 40 |
| Ilustración 15: Relación de transmisión cremallera-engranaje..... | 41 |
| Ilustración 16: Relación de transmisión entre engranajes..... | 42 |
| Ilustración 17: Posición 1 pieza en forma de "L" | 43 |
| Ilustración 18: Posición 2 pieza en forma de "L" | 43 |
| Ilustración 19: Esquema general del software | 44 |
| Ilustración 20: Lazo de control Motor C | 45 |
| Ilustración 21: Lazo de control Motor A | 46 |
| Ilustración 22: Control del balancín | 47 |
| Ilustración 23: Lazo cerrado motores grandes..... | 48 |
| Ilustración 24: Lazo cerrado final motores grandes..... | 49 |
| Ilustración 25: Lazo cerrado motor mediano..... | 50 |
| Ilustración 26: Bloque "cálculo de ángulos a partir del punto". | 50 |
| Ilustración 27: Mecanismo del robot. | 51 |
| Ilustración 28: Posibles ensamblajes del mecanismo | 53 |
| Ilustración 29: Bloque "Comparador". | 56 |
| Ilustración 30: Comportamiento de la memoria. | 56 |
| Ilustración 31: Bloque "Obtención de coordenadas"..... | 57 |
| Ilustración 32: Visualización gráfica d ela solución..... | 61 |
| Ilustración 33: Bloque "datos a escribir". | 62 |
| Ilustración 34: Conexión Simulink 1..... | 65 |
| Ilustración 35: Conexión Simulink 2..... | 65 |
| Ilustración 36: Referencia en el folio. | 66 |
| Ilustración 37: Ubicación del robot. | 66 |
| Ilustración 38: Sudoku a resolver. | 67 |
| Ilustración 39: Sudoku resuelto por el robot..... | 67 |
| Ilustración 40: Esquema general Simulink..... | 73 |
| Ilustración 41: Esquema Simulink "Obtención de ángulos y control motores"..... | 74 |
| Ilustración 42: Esquema "Simulink motor C". | 75 |
| Ilustración 43: Esquema Simulink "motor A". | 75 |
| Ilustración 44: Esquema Simulink "motor B" | 75 |
| Ilustración 45: Diseño del número 1..... | 85 |
| Ilustración 46: Diseño del número 2..... | 85 |
| Ilustración 47: Diseño del número 3..... | 86 |
| Ilustración 48: Diseño del número 4..... | 86 |
| Ilustración 49: Diseño del número 5..... | 87 |
| Ilustración 50: Diseño del número 6..... | 87 |
| Ilustración 51: Diseño del número 7..... | 88 |
| Ilustración 52: Diseño del número 8..... | 88 |
| Ilustración 53: Diseño del número 9..... | 89 |

| | |
|--|-----|
| Ilustración 54: Montaje. Paso 1..... | 92 |
| Ilustración 55: Montaje. Paso 2..... | 92 |
| Ilustración 56: Montaje. Paso 3..... | 93 |
| Ilustración 57: Montaje. Paso 4..... | 93 |
| Ilustración 58: Montaje. Paso 5..... | 94 |
| Ilustración 59: Montaje. Paso 6..... | 94 |
| Ilustración 60: Montaje. Paso 7..... | 95 |
| Ilustración 61: Montaje. Paso 8..... | 95 |
| Ilustración 62: Montaje. Paso 9..... | 96 |
| Ilustración 63: Montaje. Paso 10..... | 96 |
| Ilustración 64: Montaje. Paso 11..... | 97 |
| Ilustración 65: Montaje. Paso 12..... | 97 |
| Ilustración 66: Montaje. Paso 13..... | 98 |
| Ilustración 67: Montaje. Paso 14..... | 98 |
| Ilustración 68: Montaje. Paso 15..... | 99 |
| Ilustración 69: Montaje. Paso 16..... | 99 |
| Ilustración 70: Montaje. Paso 17..... | 100 |
| Ilustración 71: Montaje. Paso 18..... | 100 |
| Ilustración 72: Montaje. Paso 19..... | 101 |
| Ilustración 73: Montaje. Paso 20..... | 101 |
| Ilustración 74: Montaje. Paso 21..... | 102 |
| Ilustración 75: Montaje. Paso 22..... | 102 |
| Ilustración 76: Montaje. Paso 23..... | 103 |
| Ilustración 77: Montaje. Paso 24..... | 103 |
| Ilustración 78: Montaje. Paso 25..... | 104 |
| Ilustración 79: Montaje. Paso 26..... | 104 |
| Ilustración 80: Montaje. Paso 27..... | 105 |
| Ilustración 81: Montaje. Paso 28..... | 105 |
| Ilustración 82: Montaje. Paso 29..... | 106 |
| Ilustración 83: Montaje. Paso 30..... | 106 |
| Ilustración 84: Montaje. Paso 31..... | 107 |
| Ilustración 85: Montaje. Paso 32..... | 107 |
| Ilustración 86: Montaje. Paso33..... | 108 |
| Ilustración 87: Montaje. Paso 34..... | 108 |
| Ilustración 88: Montaje. Paso 35..... | 109 |
| Ilustración 89: Montaje. Paso 36..... | 109 |
| Ilustración 90: Montaje. Paso 37..... | 110 |
| Ilustración 91: Montaje. Paso 38..... | 110 |
| Ilustración 92: Montaje. Paso 39..... | 111 |
| Ilustración 93: Montaje. Paso 40..... | 111 |
| Ilustración 94: Montaje. Paso 41..... | 112 |
| Ilustración 95: Montaje. Paso 42..... | 112 |
| Ilustración 96: Montaje. Paso 43..... | 113 |
| Ilustración 97: Montaje. Paso 44..... | 113 |
| Ilustración 98: Montaje. Paso 45..... | 114 |
| Ilustración 99: Montaje. Paso 46..... | 114 |
| Ilustración 100: Montaje. Paso 47..... | 115 |
| Ilustración 101: Montaje. Paso 48..... | 115 |
| Ilustración 102: Montaje. Paso 49..... | 116 |
| Ilustración 103: Montaje. Paso 50..... | 116 |
| Ilustración 104: Montaje. Paso 51..... | 117 |
| Ilustración 105: Montaje. Paso 52..... | 117 |
| Ilustración 106: Montaje. Paso 53..... | 118 |
| Ilustración 107: Montaje. Paso 54..... | 118 |

3. ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1: Longitud de los eslabones | 36 |
| Tabla 2: Características de las piezas impresas | 41 |
| Tabla 3: Puertos de comunicación..... | 63 |





**UNIVERSIDAD
DE LA RIOJA**

Documento 2:

MEMORIA

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

ÍNDICE DE LA MEMORIA

| | |
|---|----|
| 1. INTRODUCCIÓN | 17 |
| 2. OBJETO..... | 18 |
| 3. ALCANCE..... | 20 |
| 4. ANTECEDENTES..... | 21 |
| 5. ABREVIATURAS Y DEFINICIONES..... | 23 |
| 6. NORMAS Y REFERENCIAS | 24 |
| 6.1. Disposiciones legales y normativa aplicada..... | 24 |
| 6.2. Programas utilizados..... | 24 |
| 7. BIBLIOGRAFÍA | 25 |
| 7.1. Enlaces web | 25 |
| 7.2. Libros bibliográficos..... | 25 |
| 8. ANÁLISIS DE POSIBLES SOLUCIONES | 26 |
| 8.1. Punto de vista Hardware..... | 26 |
| 8.2. Punto de vista Software | 26 |
| 9. SOLUCIÓN ELEGIDA | 28 |
| 10. RESOLUCIÓN DEL SUDOKU | 29 |
| 10.1. Introducción histórica | 29 |
| 10.2. Reglas para la resolución | 29 |
| 10.3. Algoritmo utilizado para la resolución | 30 |
| 10.4. Diagrama de flujo de la resolución | 32 |
| 10.5. Ejemplo de resolución | 33 |
| 11. DIMENSIONES FÍSICAS..... | 35 |
| 11.1. Dimensiones de los brazos del robot | 35 |
| 11.2. Dimensiones del Sudoku | 36 |
| 11.3. Dimensiones de los números | 36 |
| 11.4. Coordenadas del origen de las casillas..... | 37 |
| 12. HARDWARE..... | 38 |
| 12.1. Componentes del kit LEGO Mindstorms Education EV3 utilizados..... | 38 |
| 12.2. Diseño e impresión 3D | 40 |
| 12.3. Relaciones de transmisión | 41 |
| 13. SOFTWARE | 44 |
| 13.1. Esquema general..... | 44 |
| 13.2. Lazos de control de los motores | 44 |
| 13.3. Obtención de las plantas y controladores de los motores | 48 |
| 13.4. Cálculo de ángulos a partir del punto | 50 |
| 13.5. Error de aproximación al punto | 56 |
| 13.6. Obtención de coordenadas..... | 57 |
| 13.7. Obtención del número y casilla a partir del sudoku solución | 61 |

| | |
|---|----|
| 14. COMUNICACIÓN | 63 |
| 14.1. Puertos utilizados..... | 63 |
| 14.2. Tipos de conexión..... | 63 |
| 14.3. Procedimiento de conexión inalámbrica Wi-Fi | 64 |
| 14.4. Parámetros en Simulink | 65 |
| 15. RESOLUCIÓN POR MEDIO DEL ROBOT | 66 |
| 15.1 Referencia en el folio..... | 66 |
| 15.2 Solución..... | 67 |
| 16. CONCLUSIONES | 68 |
| 17. VÍAS DE CONTINUACIÓN..... | 69 |

1. INTRODUCCIÓN

Este trabajo ha sido realizado por Ana Río Rodríguez, estudiante de cuarto curso del Grado en Ingeniería Electrónica Industrial y Automática en la Escuela Técnica Superior de Ingeniería Industrial de la Universidad de La Rioja.

La finalidad del mismo es ser presentado como Trabajo de Fin de Grado y obtener los créditos necesarios para la obtención del título de la especialidad anteriormente citada, así como aplicar en un entorno real los conocimientos adquiridos a lo largo de la titulación.

2. OBJETO

El objetivo del presente trabajo es diseñar y construir un robot capaz de resolver de forma autónoma *sudokus*, conocido juego matemático. Una vez resuelto alcanzada la solución del juego, esta deberá ser plasmada en un papel por medio de un sistema mecánico. Para lograr estos objetivos se emplearán componentes del kit LEGO Mindstorms Education EV3.

El objetivo de este proyecto una vez el robot es construido y programado es que se sucedan los siguientes acontecimientos:

- El usuario ha de introducir los datos de partida que contiene el *sudoku*, es decir, los datos a partir de los cuales se ha de encontrar la solución.
- Una vez introducidos los datos, un algoritmo, desarrollado en Matlab, resuelve el *sudoku* completo.
- La solución alcanzada es plasmada en el papel gracias al mecanismo (robot) desarrollado, que es gobernado de forma autónoma. Para lograrlo se emplean múltiples lazos de control y bloques desarrollados en Matlab-Simulink. El robot está compuesto por un mecanismo de cinco eslabones con dos grados de libertad. Este mecanismo sujeta un rotulador en el extremo, que será la herramienta que plasme la solución en el papel.

Para alcanzar este objetivo final primeramente se han alcanzado los siguientes objetivos parciales:

- Aprendizaje y dominio de la resolución del juego matemático *sudoku* y estudio de los algoritmos necesarios para la resolución de cualquier *sudoku* sin importar el grado de dificultad.
- Dominio del lenguaje Matlab para la implementación de múltiples funciones, así como de su herramienta Simulink para el desarrollo de bloques de control.
- Construcción física del robot a partir de las piezas disponibles en el kit LEGO Mindstorms Education EV3.
- Diseño e impresión 3D de piezas complementarias.
- Diseño de los lazos de control de los tres motores que gobiernan el movimiento del robot.
- Obtención a través de cálculos matemáticos de las ecuaciones que relacionan los ángulos de los motores con las coordenadas X,Y del efector, que en este caso es un rotulador.
- Definición de las trayectorias de los números a representar.
- Comunicación de datos entre el PC y el robot en tiempo real.

Para lograr todos estos objetivos deben ponerse en práctica numerosos conceptos estudiados a lo largo de la titulación:

- Diseño asistido por ordenador. Útiles para la construcción de un modelo 3D en el programa Solidworks.
- Teoría de mecanismos. Los conocimientos adquiridos en esta asignatura han sido necesarios para el cálculo de las posiciones y ángulos del mecanismo de cinco eslabones que compone el robot.
- Informática. Conocimientos necesarios para la creación del código que controla el robot y la resolución del *sudoku*.
- Ingeniería de control y Control y programación de robots. Los conocimientos estudiados en estas asignaturas han sido necesarios para desarrollar los lazos de control que gobiernan cada uno de los motores, así como para gobernar la posición final del efector.

3. ALCANCE

Este proyecto contempla todos los elementos, esquemas, algoritmos, *firmware* y diseños necesarios para la realización de un robot capaz de resolver un *sudoku* y mover un rotulador en el espacio para dibujar los números que componen la solución en su casilla correspondiente.

A continuación, se desglosan todos los puntos alcanzados e incorporados en el proyecto y que serán explicados en profundidad a lo largo del mismo:

- Diseño y montaje del robot a partir de piezas del kit de LEGO Mindstorms Education EV3.
- Programación del código Matlab de un algoritmo capaz de resolver cualquier *sudoku* con solución posible, a partir de los datos introducidos por el usuario.
- Lazos de control de los tres motores, desarrollados en Simulink, para determinar el movimiento del rotulador que plasma la solución en el papel.
- Proceso de comunicación entre el PC y el Bloque de programación EV3 con el fin de transmitir el programa y recibir datos del proceso en tiempo real.

4. ANTECEDENTES

En este proyecto se diseña un robot empleando el kit de Lego Mindstorms Education EV3, el cual incluye un Bloque de programación con un gran potencial, se pueden crear infinidad de diseños y aplicaciones.

La marca Lego creó la línea de productos Lego Mindstorms en septiembre de 1998, esta se enfocó como una línea de robótica para niños de entre 10 y 14 años. Posee elementos básicos de las teorías robóticas, como la unión de piezas y la programación de acciones de forma muy didáctica. El nombre del primer robot de Lego comercializado fue Robotic Invention System, en español Sistema de Invención Robotizado (RIS).

Lego Mindstorms puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por ordenador. Prácticamente todo puede ser representado en pequeña escala con las piezas de Lego Mindstorms, tal como en la vida real, como robots industriales.

LEGO Mindstorms es en la actualidad el material de construcción más eficaz para comenzar a experimentar con robots y focalizarse en el aspecto académico del aprendizaje. Hasta 2018 ha habido tres generaciones de Lego Mindstorms: el bloque RCX, el bloque NXT y el EV3. El modelo utilizado en este proyecto es el EV3, el más novedoso de ellos.

Algunos ejemplos de proyectos realizados con los diversos kits de Lego Mindstorms son:

- Robot capaz de resolver cubos de Rubik: Este proyecto creado y desarrollado con la versión NXT de Lego Mindstorms consiste en un robot que resuelve cubos de Rubik de forma autónoma con la ayuda de un sensor de color, múltiples motores que controlan el giro del cubo y varios topes para hacer girar solo una cara.

<https://www.youtube.com/watch?v=dreTvumjNyw>

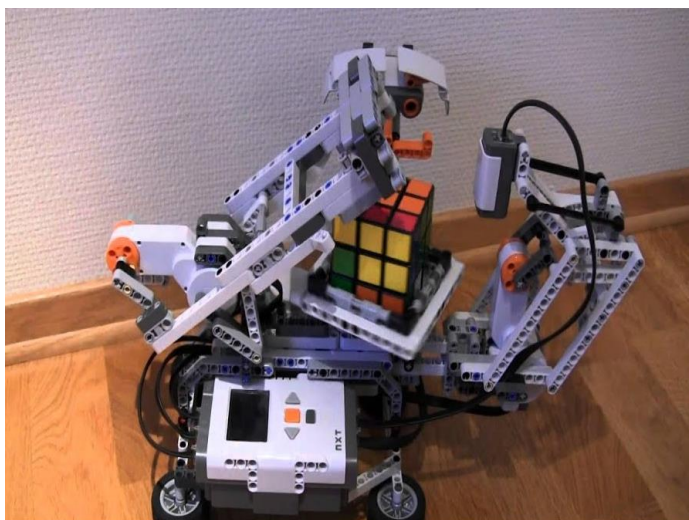


Ilustración 1: Robot resuelve cubos de Rubik

- Robot capaz de tocar una guitarra: Este robot desarrollado con el kit EV3 está compuesto por dos módulos que actúan de forma conjunta. Uno de ellos ubicado en el mástil de la guitarra y encargado de presionar la cuerda indicada, el otro módulo se encuentra en la caja de resonancia de la guitarra y hará vibrar la cuerda deseada en el momento exacto.

<https://www.youtube.com/watch?v=cXgB3IIvPHI&t=131s>

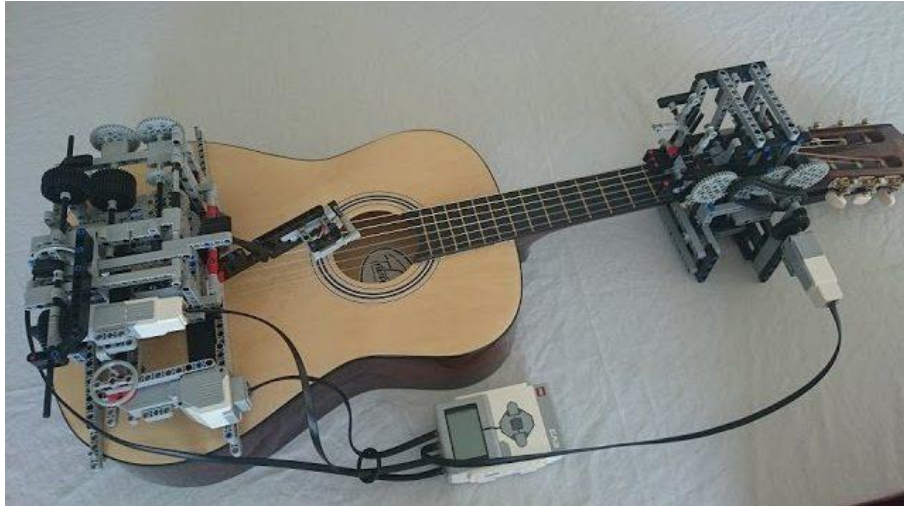


Ilustración 2: Robot capaz de tocar la guitarra

- Reloj digital: Este proyecto desarrollado con el kit NXT consiste en la construcción de un reloj digital con piezas blancas y negras acopladas a diferentes ejes que rotan simultáneamente, formando así los números correspondientes a la hora que se quiere reflejar.

https://www.youtube.com/watch?v=A_ma72r3ZiQ



Ilustración 3: Robot reloj digital

5. ABREVIATURAS Y DEFINICIONES

- Lego Mindstorms Education: Plataforma de hardware de bajo coste con un Bloque programable utilizaba en este proyecto para construir el robot.
- RCX: Robotic Command eXplorers (eXploradores de Comandos Robóticos) Siglas de la primera generación de Lego Mindstorms.
- NXT: NeXT (siguiente). Siglas de la primera generación de Lego Mindstorms.
- EV3: EVolution 3 (Evolución 3). Siglas de la tercera generación de Lego Mindstorms.
- 3D: 3 dimensiones.
- TFG: Siglas de Trabajo Fin de Grado
- Sudoku: Juego matemático que consiste en rellenar una cuadrícula de 9x9 con números del 1 al 9 sin repetir el mismo número en la fila, columna o cuadrado de 3x3 en el que se ubica.
- PC: Personal Computer. Ordenador tanto de torre como portátil.
- SD: Secure Digital. Tarjeta de memoria portátil.
- L1, L2, L3, L4 y L5: Nombre de los diferentes eslabones, o, de ser usado en una ecuación, longitud del eslabón indicado.
- Ts: Tiempo entre cada muestreo del ordenador sobre el estado del robot.
- θ : Letra para denominar ángulos

6. NORMAS Y REFERENCIAS

6.1. Disposiciones legales y normativa aplicada

- UNE 157001:2014: Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico.
- UNE 21302: Vocabulario Electrotécnico.

6.2. Programas utilizados

- Matlab: Sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio, lenguaje M.
- Simulink: Es una toolbox (herramienta) especial de Matlab que sirve para simular el comportamiento de los sistemas dinámicos. Es un entorno de programación gráfico que además permite trabajar con hardware externo.
- AutoCAD: Es un programa de dibujo asistido por ordenador (CAD) utilizado para modelado de diseños en 2 y 3 dimensiones.
- Solidworks: Es un programa de dibujo asistido por ordenador (CAD) utilizado para modelado de diseños y ensamblajes en 3 dimensiones, y planos en 2 dimensiones.

7. BIBLIOGRAFÍA

7.1. Enlaces web

Componentes del kit Lego EV3, características de los mismos e historia de la marca Lego Mindstorms:

- Página oficial de Lego Mindstorms. Recuperado de:
<https://www.lego.com/es-es/mindstorms>
- Lego Mindstorms (2019). Recuperado de:
https://es.wikipedia.org/wiki/Lego_Mindstorms

Ejemplos de proyectos de Lego Mindstorms:

- Instrucciones de montaje LEGO Mindstorms EV3 (2018). Recuperado de:
<http://juegosrobotica.es/instrucciones-lego-mindstorms-ev3/>

Metodologías de resolución de sudokus:

- Métodos de resolución. Recuperado de:
<http://www.sudokumania.com.ar/metodos>
- Técnicas de Resolución de Sudokus. Recuperado de:
<http://www.playsudoku.biz/tecnicas-solucion-sudokus>

Ejemplos de código y funciones de y Simulink:

- Página oficial de Mathworks. Recuperado de:
<https://la.mathworks.com/academia/examples.html>

7.2. Libros bibliográficos

- Henao López, Juan Carlos. (2018). Manual de manejo de dispositivos Lego Mindstorms: un apoyo para la enseñanza de las matemáticas. *Pereira*.
- Ogata, Katsushiko. (2003). Ingeniería de Control Moderna. *Pearson Educación*.

8. ANÁLISIS DE POSIBLES SOLUCIONES

En este proyecto se plantearon múltiples alternativas antes de elegir la solución definitiva. Todas ellas emplean para el desarrollo del robot (mecanismo empleado para transcribir la solución) el kit LEGO Mindstorms Education EV3, ya que es una plataforma que la Universidad de La Rioja tiene en su poder y se adapta perfectamente al objeto del trabajo.

Debido a que este proyecto se compone de una parte Hardware y otra Software se plantearon dos grupos muy diferenciados de opciones, las relacionadas con el aspecto físico del robot y los mecanismos utilizados para plasmar la solución en el papel y las opciones enfocadas a la programación del robot.

8.1. Punto de vista Hardware

A la hora de planear el modo de plasmar la solución en el papel se estudiaron varias opciones:

- Impresora: Esta opción consiste en construir una especie de impresora con las piezas de LEGO. En este caso se introduce el papel con el modelo de *sudoku* por un extremo, una serie de rodillos hacen que este papel avance o retroceda y se deslice hacia los lados. Por último, un rotulador, cuyo único movimiento es lineal de manera perpendicular al papel, sube y baja dejando la solución plasmada.
- Brazos articulados: Esta solución consiste en dejar el papel fijo, así como el bloque del robot y que sea un mecanismo compuesto por dos brazos articulados y unidos en el extremo el que realice el movimiento. En la unión de los dos brazos se sujeta el rotulador encargado de moverse en el espacio (X,Y,Z) y plasmar la solución.
- Robot con ruedas: La última opción barajada consiste en dejar el papel fijo, y construir un robot con ruedas que pueda desplazarse en el plano. Los ejes X e Y serían controlados desde el robot y el rotulador simplemente sube y baja según se quiera dibujar o no.

8.2. Punto de vista Software

La programación de la resolución del *sudoku* puede ser implementada en diversos lenguajes y programas, pero se debe tener en cuenta que posteriormente el programa se ha de poder comunicar con el bloque de programación de Lego.

- Programación en Python: Para poder programar un Bloque de programación de Lego con Python primero se ha de descargar y flashear una imagen de Python en una tarjeta micro SD e introducir dicha tarjeta en la ranura SD del Bloque EV3. Posteriormente, instalar el programa Visual Studio Code en el PC y por último descargar el driver correspondiente para Lego Education EV3. Python es una plataforma muy potente a la hora de crear códigos complejos.

- Programación en Lego Mindstorms EV3: Lego ofrece su propia plataforma de programación. Es una plataforma enfocada a niños, con un interfaz muy fácil y didáctico. No deja por ello de tener infinidad de posibilidades, pero es más complicado agregar código complejo como puede ser la resolución de un *sudoku*.
- Programación con Matlab: Matlab ofrece un amplio abanico de opciones y es un programa muy potente en cuanto a la complejidad del código que acepta. Su herramienta Simulink permite la integración de bloques de control. Para poder programar el Bloque EV3 simplemente se ha de descargar e instalar la librería de Lego Mindstorms EV3 compatible con Matlab.

9. SOLUCIÓN ELEGIDA

Todas las opciones de hardware son compatibles con todas las soluciones software planteadas anteriormente, por lo tanto, se ha optado por seleccionar la solución hardware y software de forma independiente. La solución finalmente elegida es la combinación óptima de todas las opciones posibles.

En cuanto al hardware, se ha decidido emplear un robot plano paralelo 2RRR. En esta configuración, el robot presenta una estructura de dos brazos unidos en el extremo que sujeta el rotulador, cerrando así un mecanismo de cinco eslabones con dos grados de libertad. Los dos grados de libertad son controlados por dos motores independientes que determinan los ángulos de desplazamiento de los eslabones.

Todo el conjunto de escritura se comporta como un balancín, cuyo punto de apoyo es el bloque de programación que permanece fijo, al igual que el papel en el que se escribe la solución. El elemento encargado de subir y bajar el conjunto de escritura es un tercer motor rotativo.

En cuanto a la programación, todo el proyecto se desarrolla en Matlab y su herramienta Simulink. La resolución del *sudoku* se desarrolla con código Matlab, así como otras muchas funciones necesarias para el correcto funcionamiento del conjunto y que serán explicadas más adelante en este documento.

En Simulink se desarrollan todos los lazos de control, se unen los bloques de funciones y se incorporan los elementos de LEGO EV3 como motores y sensores.

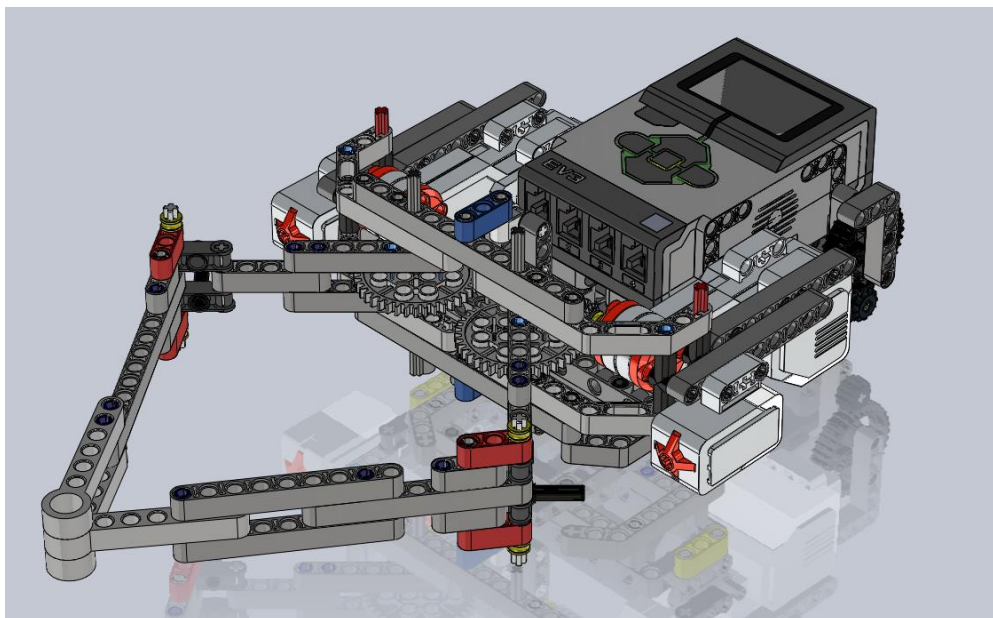


Ilustración 4: Modelo 3D del robot complete

10. RESOLUCIÓN DEL SUDOKU

10.1. Introducción histórica

El *sudoku*, es un juego matemático que se publicó por primera vez a finales de la década de 1970 y se popularizó en Japón en 1986. Mucho antes, en el siglo XVIII, el matemático suizo, Leonhard Euler de Basilea (1707-1783), creó un sistema de probabilidades para representar una serie de números sin repetir. Debido a este estudio se considera a Leonhard Euler de Basilea como inventor de este juego.

Posteriormente, en 1970 la editorial *Math Puzzles and Logic Problems* publicaba una sección llamada *Number place*, es la primera vez que este juego matemático aparece como un pasatiempo, aunque años más tarde se perdió en el olvido.

En 1984 el periódico japonés *Monthly Nikolist* publicó una sección de pasatiempos llamada *Sūji wa dokushin ni kagiru* (“los números deben estar solos”). Fue Kaji Maki, presidente de *Nikoli*, quien le puso el nombre. El nombre se abrevió a *sūdoku* (sū = número, doku = solo).

Este juego matemático, ya bautizado como *sudoku*, se dio a conocer internacionalmente en 2005, cuando numerosos periódicos empezaron a publicarlo en su sección de pasatiempos.

10.2. Reglas para la resolución

El *sudoku* está formado por una cuadrícula de dimensiones 9x9, es decir, 81 casillas en total. A su vez esta cuadrícula está subdividida en 9 cuadrados, formados por grupos de 3x3 casillas.

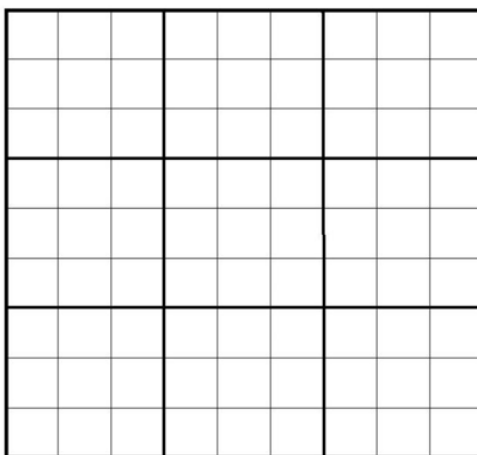


Ilustración 5: Cuadrícula del Sudoku

El objetivo de este juego es completar todas las casillas, cada una con un número del 1 al 9 y las reglas para hacerlo son:

- No debe repetirse ningún número en la misma fila
- No debe repetirse ningún número en la misma columna
- No debe repetirse ningún número en el mismo cuadrado de 3x3 casillas

Para su resolución, los *sudokus* vienen parcialmente resueltos, no se parte de un *sudoku* vacío, sino que algunas casillas ya cuentan con su número correspondiente y a partir de ellos se ha de buscar la solución. A menor cantidad de números de inicio, mayor será la dificultad de resolución del *sudoku*.

10.3. Algoritmo utilizado para la resolución

Para la resolución del *sudoku* se ha utilizado un algoritmo desarrollado en Matlab, con el fin de posteriormente comunicar los datos a Simulink y que sea el robot quien transfiera los resultados al papel.

El algoritmo desarrollado es capaz de resolver cualquier *sudoku* que cuente con unos números de partida suficientes y coherentes, de tal manera que el juego tenga una solución posible.

Antes de poder resolver el *sudoku*, es necesario introducirlo en el programa. El modo de introducirlo es en forma de matriz de 9x9, en las casillas que haya un número de partida se coloca dicho número, en los huecos en blanco se coloca un 0. El nombre de la variable utilizada para introducir esta matriz es SUDOKU.

Una vez el usuario ha introducido el *sudoku* a resolver, se crea una matriz llamada “huecos” llena de 0 de tamaño 9x9, del mismo tamaño que el *sudoku*. El valor de los datos de la matriz “huecos” se actualiza con el *sudoku* introducido. En aquellas casillas del *sudoku* a resolver que haya un 0, es decir, un hueco en blanco, en la matriz “huecos” se coloca un 1. El resto de casillas se mantienen como estaban.

```
for F=1:9
    for C=1:9
        if SUDOKU(F,C)==0
            huecos(F,C)=1;
        end
    end
end
```

De modo que la nueva matriz “huecos” marca con un 1 aquellas casillas en las que se debe buscar un número para la solución y con un 0 aquellas en las que hay un número de partida y por tanto, no se deben modificar.

A continuación, comienza la resolución propiamente dicha, se busca el primer hueco a resolver en la matriz “huecos” y se empieza a aplicar un algoritmo recurrente que consiste en prueba y error. En este hueco se coloca en primer lugar el número 1 y se comprueba que dicho número no se encuentre ya en la misma fila.

```
%Comprobamos que el n°"a" no se encuentre ya en esa fila
for (m=1:9)
    if (SUDOKU(F,C)==SUDOKU(F,m) && C~=m)
        valido=0; % No válido
    end
```

Posteriormente, se comprueba que no se encuentre en la misma columna:

```
%Igualmente para la columna
if (SUDOKU(F,C) == SUDOKU(m,C) && F~=m)
    valido=0; % No válido
end
```

Y, por último, que no se encuentre en el mismo cuadrado de 3x3 casillas:

```
%Se comprueba que el n°"a" no se encuentre ya en el cuadrado
C_x=floor((C-1)/3);
C_y=floor((F-1)/3);
for (p=1:3)
    for (q=1:3)
        if
            SUDOKU(F,C)==SUDOKU(3*C_y+p,3*C_x+q) && (F~=3*C_y+p | C~=3*C_x+q)
                valido=0; % No válido
            end
        end
    end
```

Si el número pasa todas las pruebas, se confirma como solución de dicha casilla y se pasa a la siguiente casilla a evaluar. Por el contrario, si el número no cumple uno de los requisitos, sin necesidad de comprobar el resto, se incrementa su valor en una unidad y se vuelve a realizar el mismo procedimiento.

En esta forma de resolver el *sudoku* se cometen errores, pero el algoritmo está preparado para ello. Si después de probar en una casilla con todos los números del 1 al 9 ninguno cumple los requisitos de solución es debido a que en alguna casilla anterior había más de una opción y se ha elegido la errónea.

En este caso el programa busca la anterior casilla resuelta e incrementa el número elegido en una unidad. Si en esta casilla también se llega al máximo sin encontrar un número viable, se comprueba la casilla anterior y así sucesivamente hasta encontrar la casilla en la que se ha cometido el error. Una vez modificada la casilla conflictiva, el algoritmo sigue su camino hasta resolver todas las casillas.

Es muy probable que el algoritmo tenga que retroceder en varias ocasiones a casillas ya resueltas por problemas de múltiple solución. Especialmente en fases tempranas, cuando aún hay muchas casillas en blanco. Este no es un problema en los PCs actuales, en los que se dispone de una velocidad de cálculo elevada.

Una vez el *sudoku* ha sido resuelto se pasan los datos de las casillas que hay que cumplimentar en el papel a Simulink para que el robot comience la escritura.

10.4. Diagrama de flujo de la resolución

A continuación, se visualiza el diagrama de flujo seguido por el algoritmo para la resolución del *sudoku*, tal y como se ha explicado en el apartado anterior.

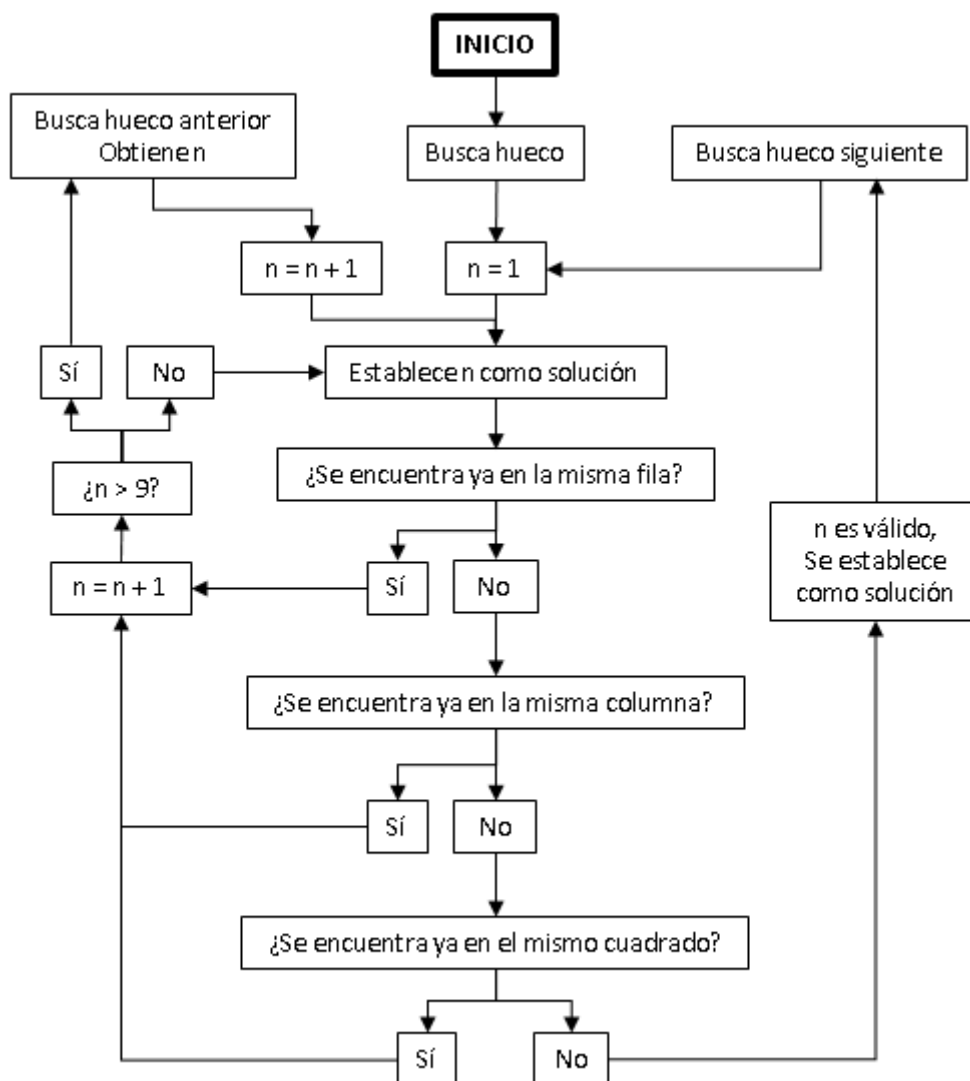


Ilustración 6: Diagrama de flujo de la resolución del Sudoku

10.5. Ejemplo de resolución

A continuación se describen los pasos necesarios para resolver un *sudoku* en el que la información de partida es la dada en la (Ilustración 7):

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | | | 7 | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | 2 | 8 | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |

Ilustración 7: Sudoku ejemplo

En primer lugar, se introduce la variable SUDOKU en forma de matriz de 9x9, con el número correspondiente en aquellas casillas que contengan un dato y un 0 en los huecos en blanco:

SUDOKU = [5 3 0 0 7 0 0 0 0; 6 0 0 1 9 5 0 0 0; 0 9 8 0 0 0 0 6 0; 8 0 0 0 6 0 0 0 3; 4 0 0 8 0 3 0 0 1; 7 0 0 0 2 0 0 0 6; 0 6 0 0 0 0 2 8 0; 0 0 4 1 9 0 0 5; 0 0 0 8 0 0 7 9]

| | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|
| SUDOKU = | | | | | | | | |
| 5 | 3 | 0 | 0 | 7 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 9 | 5 | 0 | 0 | 0 |
| 0 | 9 | 8 | 0 | 0 | 0 | 0 | 6 | 0 |
| 8 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 3 |
| 4 | 0 | 0 | 8 | 0 | 3 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 6 |
| 0 | 6 | 0 | 0 | 0 | 0 | 2 | 8 | 0 |
| 0 | 0 | 0 | 4 | 1 | 9 | 0 | 0 | 5 |
| 0 | 0 | 0 | 0 | 8 | 0 | 0 | 7 | 9 |

Ilustración 8: Matriz SUDOKU

Una vez introducido el *sudoku* a resolver, se ejecuta el programa encargado de buscará la solución. A continuación, se muestra la matriz “huecos” previamente explicada que recoge la información sobre qué casillas contienen un dato de partida y cuáles no (Ilustración 9).

| | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|
| huecos = | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

Ilustración 9: Matriz huecos

Y finalmente el programa alcanza la solución (Ilustración 10):

| | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|
| SUDOKU = | | | | | | | | |
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Ilustración 10: Solución en Matlab

Este algoritmo de resolución es bastante rápido, encuentra la solución en aproximadamente 1 segundo. Lo cual es inalcanzable para un ser humano.

11. DIMENSIONES FÍSICAS

En esta sección se presentan las dimensiones de los diferentes elementos que componen el trabajo.

11.1. Dimensiones de los brazos del robot

Como se ha descrito en secciones anteriores, el robot empleado está compuesto por un mecanismo de cinco eslabones con dos grados de libertad como el de la Ilustración 11.

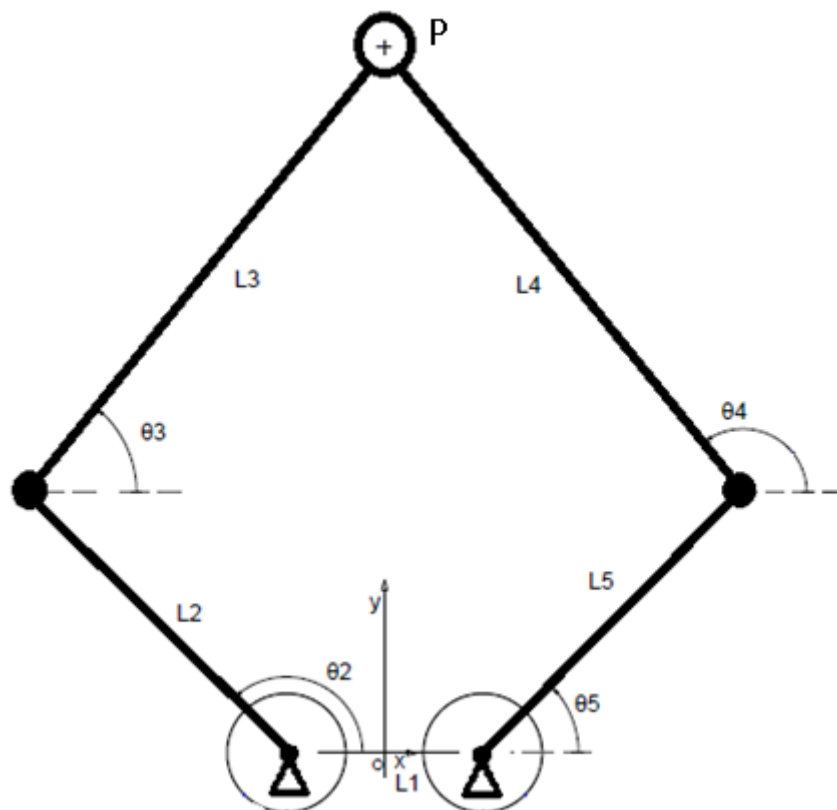


Ilustración 11: Dimensiones de los brazos del robot

Los dos grados de libertad vienen dados por los ángulos θ_2 y θ_5 los cuales serán controlados por dos motores independientes. Los eslabones L2 y L3 presentan una rótula en su unión por lo que están articulados, del mismo modo se encuentran unidos los eslabones L5 y L4.

En la unión de los eslabones L3 y L4 es donde se ubica el rotulador que refleja la solución sobre el papel, esta unión también es articulada.

Todos los eslabones tienen una longitud constante ya que no hay ningún tipo de corredera, dichas longitudes se muestran en la siguiente tabla:

Tabla 1: Longitud de los eslabones

| Eslabón | Longitud (mm) |
|---------|---------------|
| L1 | 48 |
| L2 | 90 |
| L3 | 140 |
| L4 | 140 |
| L5 | 90 |

11.2. Dimensiones del Sudoku

Un *sudoku* tiene 81 casillas, 9 filas y 9 columnas. Dado el alcance de los brazos del robot, el *sudoku* utilizado (cuadrícula en la que se transcribe la solución final) tiene unas dimensiones de 90x90 mm, es decir cada casilla tiene una dimensión de 10x10 mm.

Este tamaño de *sudoku* es lo suficientemente grande como para que los números escritos por el robot sean completamente legibles, y al mismo tiempo, lo suficientemente pequeños como para que todas y cada una de las casillas se encuentren dentro del rango de operación de los brazos del robot y estén a su alcance. (Acudir al Plano N°3)

11.3. Dimensiones de los números

Como se ha citado anteriormente cada una de las casillas tiene como dimensiones 10x10 mm. A la hora de dibujar el número deseado con el robot se quiere evitar que dicho número se salga de los límites de la casilla, es por ello que la dimensión de todos los números es de 7x7 mm teniendo así 1,5 mm de margen en cada lado (arriba, abajo, izquierda y derecha).

Para definir la trayectoria de los números se ha cogido como referencia la cuadrícula de la Ilustración 12, en la que aparece la dimensión total de la casilla y dentro de ella un cuadrado azul que delimita la dimensión máxima del número. El sistema de referencia para los ejes X e Y parte del punto de “origen” del cuadrado azul.

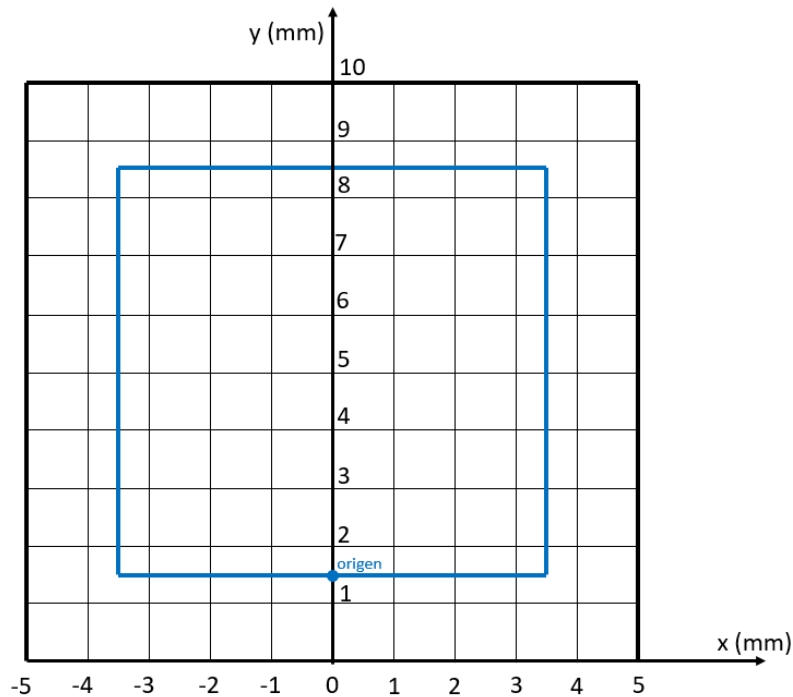


Ilustración 12: Cuadrícula dimensiones de los números.

11.4. Coordenadas del origen de las casillas

El rango de coordenadas del *sudoku* en el eje X es de -45 a 45, y en el eje Y de 110 a 200, teniendo en cuenta que estas coordenadas están expresadas en mm, se tienen las dimensiones totales del *sudoku* 90x90 mm.

Las coordenadas del origen de cada casilla (punto “origen” de la Ilustración 12) se obtienen de la siguiente manera:

- **Coordenada X:** El *sudoku* tiene 9 casillas de ancho, cada una de ellas mide 10 mm y el origen de las mismas en el eje X se encuentra en el punto medio, dejando 5mm a cada lado.
- **Coordenada Y:** La coordenada Y del origen de la casilla depende de la dimensión del número escrito en su interior. En este proyecto los números miden 7mm de altura, y se ubican de manera centrada en la casilla. Por tanto, existe un margen de 1,5mm tanto en la parte superior como inferior del número. El origen Y de la casilla se ubica 1,5mm por encima del límite inferior de la misma.

12. HARDWARE

La parte física y tangible de este proyecto, es decir el hardware, ha sido construida con dos kits de Lego Mindstorms Education EV3. Uno de ellos es denominado “set principal” el cual incluye los componentes fundamentales como el bloque de programación, los sensores y los actuadores que se explican más adelante en este documento, así como una serie de piezas que facilitan el montaje de modelos. El otro kit es denominado “kit de expansión”, este kit se compone de piezas extra como ejes, ruedas, juntas y piezas decorativas con vista a la realización de proyectos más voluminosos que requieran un número elevado de piezas, pero no más actuadores o sensores.

12.1. Componentes del kit LEGO Mindstorms Education EV3 utilizados

12.1.1. Bloque de programación

Es el centro de control y central eléctrica del robot. Este bloque de programación está en contacto continuo con el PC y es el encargado de recibir la información y mandarla al actuador indicado, y al mismo tiempo recibir los datos de los sensores y transmitirlos al PC.

- Sistema operativo: LINUX
- Controlador ARM9 de 300 MHz
- Memoria Flash: 16 MB
- RAM: 64 MB
- Resolución de la pantalla del Bloque EV3: 178x128/blanco y negro
- Comunicación USB 2.0 a equipo anfitrión: hasta 480 Mbit/s.
- Comunicación USB 1.1 a anfitrión: hasta 12 Mbit/s.
- Tarjeta Micro SD: admite SDHC, versión 2.0, 32 GB máx.
- Puertos de los sensores y del motor
- Conectores: RJ12
- Admite Identificación automática
- Alimentación: 6 baterías AA

12.1.2. Motor grande

El motor grande es el motor más potente del kit LEGO Mindstorms Education EV3. Tiene un Sensor de rotación incorporado con resolución de 1 grado para un control preciso.

- Funciona a 160–170 rpm
- Par de rotación de 20 Ncm
- Par de rotor bloqueado de 40 Ncm

En el modelo realizado en este proyecto se hace uso de dos motores grandes, cada uno de ellos se hace cargo de la rotación de los ángulos θ_2 y θ_5 a través de una transmisión cremallera-engranaje. Por tanto, estos motores gobiernan la posición del efector (rotulador).

12.1.3 Motor mediano

El motor mediano es más ligero y pequeño que el motor grande, esto hace que sea más rápido, pero menos potente. Al igual que el anterior también incluye un sensor de rotación incorporado con resolución de 1 grado.

- Funciona a 240–250 rpm
- Par de rotación de 8 Ncm
- Par de rotor bloqueado de 12 Ncm

En el modelo de este proyecto se utiliza un motor mediano, el cual se encarga de la posición Z del bolígrafo. Para ello hace uso del simple sistema de un balancín, explicado en el apartado 12.3.2.

12.1.4. Sensor táctil

El Sensor táctil es un sensor analógico que puede detectar el momento en el que se presiona y se suelta el botón rojo del sensor.

El hecho de que el sensor sea de tipo analógico significa que existen tres posibilidades de posición detectables: presionado, no presionado o en contacto (punto intermedio entre estar completamente presionado o no presionado).

En este proyecto se utilizan dos sensores táctiles diferentes, uno para cada brazo del robot. Son requeridos para determinar la posición de los mismos, en ausencia de estos sensores sería imposible controlar la posición del robot ya que no se sabría de qué punto se parte y por tanto imposible calcular la rotación necesaria de cada motor grande para alcanzar la posición deseada.

Cuando ambos brazos están presionando su sensor táctil correspondiente el robot se encuentra en la posición de reposo. La señal analógica nos informa de esta posición y se determina el valor exacto de los ángulos: $\theta_2 = 180^\circ$ y $\theta_5 = 0^\circ$.

12.2. Diseño e impresión 3D

A pesar de la amplia gama de piezas técnicas ofrecidas por LEGO para el montaje de modelos, ninguna de ellas era adecuada para la sujeción del rotulador empleado como efector.

Por este motivo se han diseñado e impreso en 3D, tres piezas (dos de ellas iguales) específicamente para este uso.

- Pieza sujeción rotulador 1: La pieza desarrollada presenta un orificio de 12 mm de diámetro en el que se inserta la “Pieza sujeción rotulador 2”. Para fijarse a los brazos del robot dispone de 6 orificios de 5 mm de diámetro con un rebaje de 0,5mm para encajar a la perfección con los conectores técnicos oficiales de Lego Mindstorms.

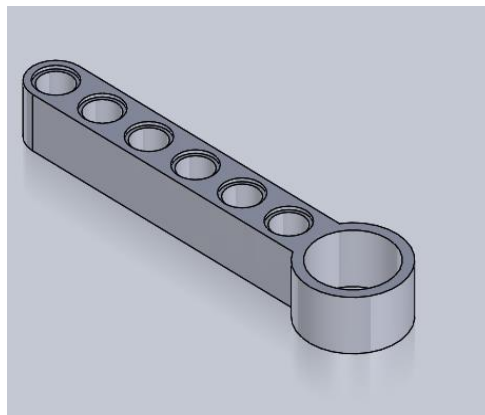


Ilustración 13: Pieza sujeción rotulador 1.

- Pieza sujeción rotulador 2: Esta pieza es muy semejante a la “Pieza sujeción rotulador 1”. Presenta un orificio de 10mm de diámetro en el que se introduce el rotulador, este orificio se prolonga 17mm con una pared de 1mm de espesor formando un aro en el que se introducen dos “Piezas sujeción rotulador 1”.

Esta pieza cuenta con los mismos orificios de fijación al robot que la pieza anterior.

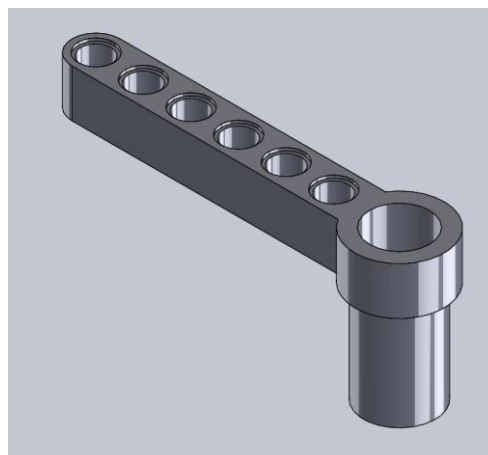


Ilustración 14: Pieza sujeción rotulador 2.

Las piezas han sido diseñadas en Solidworks (planos N° 1 y N° 2 del documento Planos) e impresas con plástico PLA blanco.

Tabla 2: Características de las piezas impresas

| Nombre | Volumen (cm ³) | Peso (g) | Cantidad |
|----------------------------|----------------------------|----------|----------|
| Pieza sujeción rotulador 1 | 3,67 | 4,62 | 2 |
| Pieza sujeción rotulador 2 | 4,54 | 5,72 | 1 |

12.3. Relaciones de transmisión

12.3.1 Brazos

La dinámica de los brazos es controlada por dos motores grandes del kit Lego EV3, uno se encarga del ángulo θ_2 y otro del ángulo θ_5 . Pero la rotación de los motores no es directamente la rotación de dichos ángulos, en realidad, existe una relación de transmisión entre la cremallera que rodea el eje de cada motor y los engranajes de 40 dientes acoplados a los eslabones L2 y L5 de los brazos del robot. Ilustración 15.

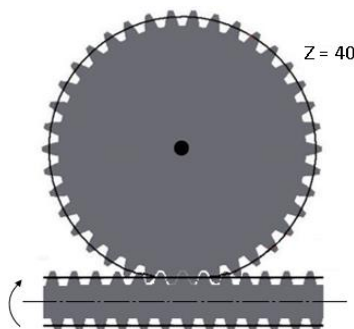


Ilustración 15: Relación de transmisión cremallera-engranaje.

Por cada vuelta del motor, el engranaje rota 1 diente, lo que equivale a $\frac{360^\circ}{40} = 9^\circ$. Por lo que la relación de transmisión, en función de los ángulos, es:

$$i_{\text{motor grande}} = \frac{9^\circ}{360^\circ} = 0,025 \quad (\text{Ec.1})$$

12.3.2. Balancín

A la hora de desplazar el lápiz de un punto a otro, hay momentos en los que se quiere apoyar el boli sobre el papel y escribir y otros en los que solamente se quiere cambiar de posición sin dejar rastro sobre el papel.

Esto se controla con el balancín, el cual domina el eje Z del sistema y depende del motor de rotación mediano del kit Lego EV3.

Este balancín se ubica en la parte trasera de un lateral del robot y está compuesto por dos engranajes: uno de ellos de 12 dientes, acoplado directamente al motor mediano, y el otro de 36 dientes colocado en transmisión con el anterior. El segundo engranaje cuenta con una pieza acoplada, en forma de “L” que será la encargada de levantar y bajar el conjunto delantero del robot.

La relación de transmisión entre los dos engranajes, en función del número de dientes, es, por tanto:

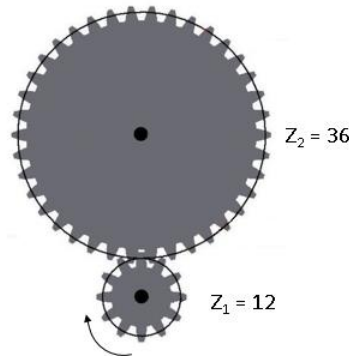


Ilustración 16: Relación de transmisión entre engranajes.

$$i_{\text{motor mediano}} = \frac{Z_1}{Z_2} = \frac{12}{36} = 0,3 \quad (\text{Ec.2})$$

La posición de reposo desde la cual parte el sistema del balancín es la posición 1 (Ilustración 17) cuando la pieza en forma de “L” se encuentre en dicha posición significa que el sistema ha rotado 0° respecto al punto de referencia, porque se encuentra, precisamente, en el punto de referencia.

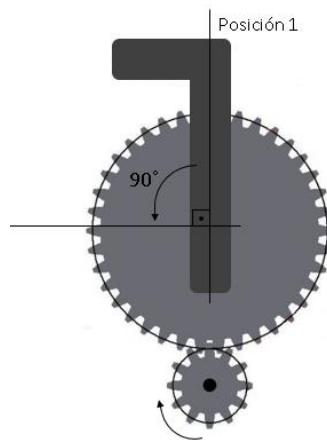


Ilustración 17: Posición 1 pieza en forma de "L"

Cuando el balancín se encuentra en la posición 1, no está ejerciendo ninguna fuerza sobre el conjunto de escritura delantero, por lo que, al dejarlo actuar libremente, el bolígrafo está en contacto con el papel y por tanto el robot está en modo “escritura”.

Por otro lado, se tiene la posición 2 (Ilustración 18) En esta posición la pieza en forma de “L” ha rotado 60° en sentido positivo, ya que lo ha hecho en sentido contrario a las agujas del reloj, respecto a la posición de reposo (Posición 1).

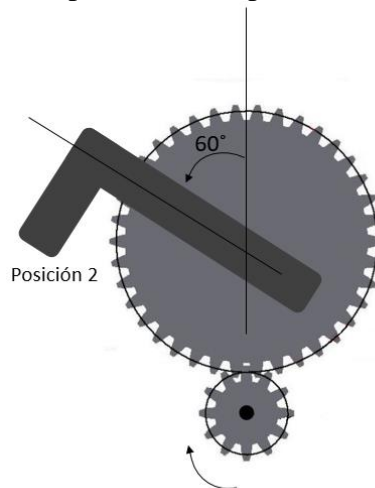


Ilustración 18: Posición 2 pieza en forma de "L"

Cuando el balancín se encuentra en la posición 2, está ejerciendo una fuerza sobre el conjunto de escritura delantero superior a su peso, provocando que el bolígrafo se separe del papel y que el robot se encuentre en modo “no escritura”.

13. SOFTWARE

13.1. Esquema general

El esquema general que presenta el software de este proyecto es el siguiente:

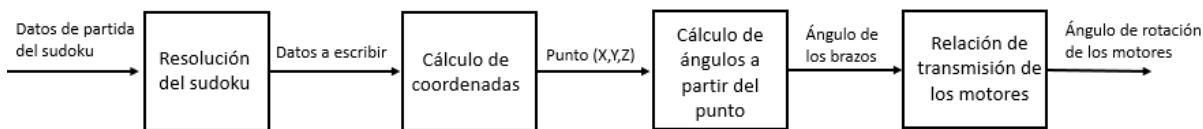


Ilustración 19: Esquema general del software

El usuario introduce los datos de partida del *sudoku* y como punto final se requiere el ángulo de rotación de cada uno de los motores para escribir la solución obtenida.

Como pasos intermedios primero se tiene que resolver el *sudoku*, posteriormente pasar estos datos a escribir a un bloque que determine las coordenadas del punto que se ha de escribir en cada momento. A continuación, otro bloque pasará estas coordenadas a ángulos de posición de los dos brazos, y, por último, con ayuda de las relaciones de transmisión explicadas anteriormente, se obtienen los ángulos de rotación de los tres motores en cada momento.

Todo el software del proyecto se explica con detenimiento y profundidad en los siguientes apartados.

13.2. Lazos de control de los motores

13.2.1 Motor grande 1

Como ya se ha explicado anteriormente el robot cuenta con dos motores grandes cada uno de ellos encargado de uno de los engranajes de 40 dientes y por tanto de los ángulos θ_2 y θ_5 .

En este apartado se explica el motor grande 1 el cual domina el ángulo θ_2 . Este motor está conectado al puerto C de salida del bloque de programación, y su punto de referencia será controlado por el sensor táctil ubicado en el puerto 4 de entradas de dicho bloque.

En la siguiente imagen (Ilustración 20) se encuentra el lazo de control de este motor y se explica en detalle a continuación.

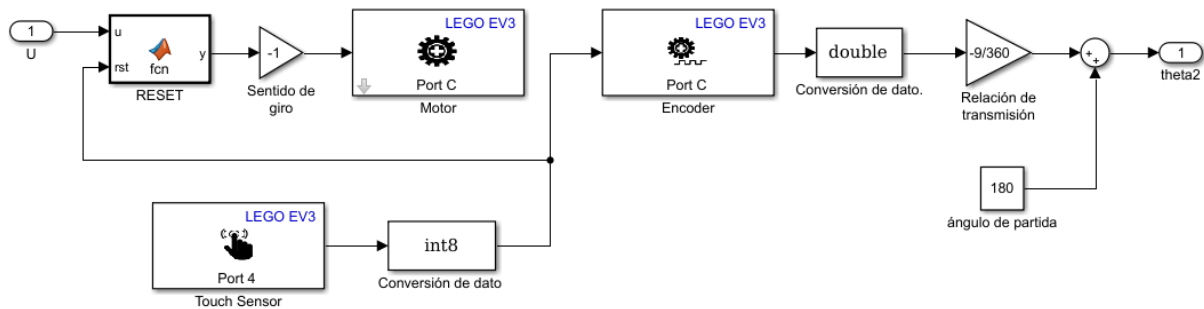


Ilustración 20: Lazo de control Motor C

Si a este motor se le aplica una potencia positiva (U) la cremallera acoplada al eje gira en sentido horario, y dada la transmisión cremallera-engranaje, este último, gira también en sentido horario.

Un ángulo está orientado en sentido positivo cuando el giro desde la semirrecta origen a la semirrecta extremo se realiza en sentido antihorario. Es por ello que se ha de colocar una ganancia -1 para que el motor invierta su sentido de giro y de esta manera el ángulo rotado por el engranaje sea positivo, cuando la potencia de entrada sea positiva.

Con la ayuda del encoder de precisión de 1° incorporado en el motor, se controla la rotación de la cremallera. Esta rotación será negativa cuando el giro sea antihorario, pero el sentido antihorario es positivo en cuanto a ángulos. Por ello se debe colocar nuevamente una ganancia negativa para invertir el signo. Con el fin de transformar los grados de rotación de la cremallera en grados de rotación del engranaje, se multiplica por $9/360$ dada la relación de transmisión descrita en el apartado 12.3. La ganancia total teniendo en cuenta estas dos condiciones es $-9/360$, para así obtener el ángulo rotado por θ_2 .

Para realizar estas operaciones es necesario modificar el tipo de dato para que sea de tipo *double*, ya que en caso contrario Simulink no acepta la operación debido a la incompatibilidad de los tipos de datos.

Cuando el robot comienza a ejecutar el programa, lo primero que hace es activar el motor para llevar el brazo a la posición de partida. Esta posición se alcanza cuando el sensor táctil es pulsado por el brazo. La posición de partida de θ_2 es 180° , este ángulo es sumado al final de todos los cálculos ya que cuando el sensor táctil se activa resetea el valor del ángulo a 0° .

En el anexo 2.6. se ve el código de la función RESET que simplemente activa el motor con una potencia de 50% respecto al máximo, cuando el reset no ha sido realizado previamente. Esta potencia es positiva porque el engranaje ha de girar en sentido antihorario para alcanzar la posición de partida.

En caso de ya haberse reseteado, deja pasar el valor de U sin modificarlo en absoluto.

13.2.2. Motor grande 2

De manera semejante al Motor grande 1, este Motor grande 2 está asociado al ángulo θ_5 .

Se encuentra conectado al puerto A de salida del bloque de programación y su posición de partida es controlada por el puerto 1 de entradas.

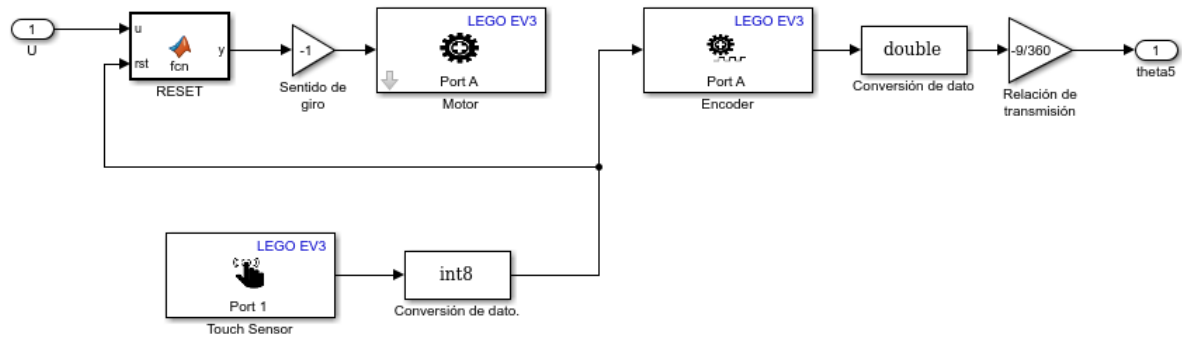


Ilustración 21: Lazo de control Motor A

Este conjunto se encuentra en modo espejo respecto al conjunto del Motor grande 1, es por ello que los sentidos de rotación tanto de cremallera y ángulo siguen el mismo comportamiento y también requieren de ganancias negativas para subsanar los criterios de referencia.

Al tratarse de un sistema idéntico en cuanto a montaje, la relación de transmisión también es 9/360.

Solo hay dos diferencias respecto al Motor grande 1. Una de ellas es que el ángulo θ_5 tiene un valor de 0° cuando el brazo se encuentra en la posición de partida, tocando el sensor táctil. Por lo que no es necesario sumar ningún valor al ángulo tras el reset.

La otra diferencia es que la potencia dada al motor para resetear es de -50%, este símbolo negativo es necesario ya que el engranaje ha de girar en sentido horario para alcanzar la posición de partida. (anexo 2.7.)

13.2.3. Motor mediano

El Motor mediano, encargado del sistema de balancín del robot, es el que cuenta con un control más simple ya que no hay ningún sensor que resetee su posición.

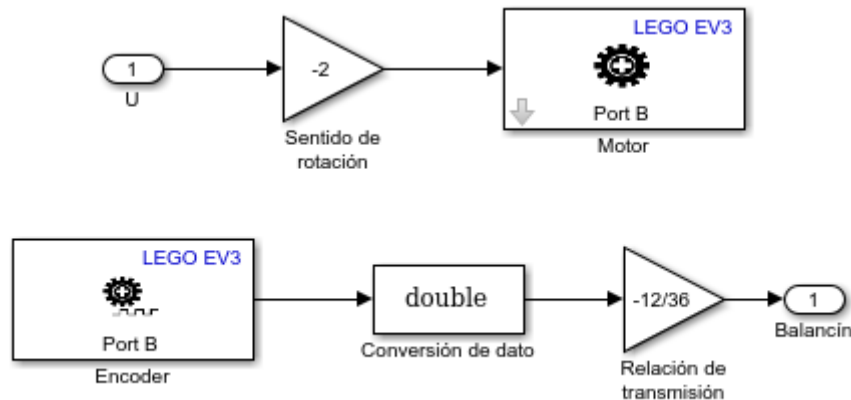


Ilustración 22: Control del balancín

La potencia de entrada (U) es multiplicada por 2 debido a que necesita una potencia mayor para poder contrarrestar el peso del conjunto de escritura delantero del robot.

El encoder incorporado con precisión de 1° ayuda a saber en todo momento el ángulo del balancín, pero previamente se ha de multiplicar por la relación de transmisión, que en este caso es $12/36$.

Cabe destacar que, al no contar con un sensor para el reset de este motor, cada vez que se transmita el programa al robot para su ejecución, la pieza en forma de “L” ha de estar colocada verticalmente, en su posición de partida. En caso contrario el robot pierde la referencia y realmente la pieza en forma de “L” se encuentra en una posición diferente a la que el robot está manejando.

13.3. Obtención de las plantas y controladores de los motores

13.3.1. Motores grandes

Dado que ambos motores grandes son iguales, presentan las mismas características y por tanto los siguientes cálculos pueden ser aplicados a los dos por igual. El tipo de controlador utilizado es proporcional, es decir, de tipo P.

El lazo cerrado utilizado para los motores grandes presenta el siguiente esquema:

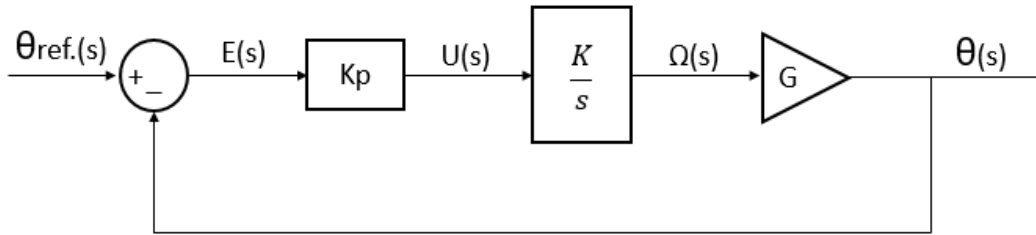


Ilustración 23: Lazo cerrado motores grandes.

Dado que el motor grande tiene una velocidad angular máxima de 170 rpm, su función de transferencia puede aproximarse por:

$$170 \text{ rpm} \rightarrow 170 \cdot \frac{360^\circ}{60 \text{ s}} = 10,2 \text{ } \theta/\text{s} \quad (\text{Ec.3})$$

$$\frac{\Omega(s)}{U(s)} = K = 10,2 \quad (\text{Ec.4})$$

$$\frac{\theta}{U(s)} = \frac{K}{s} \cdot \frac{9}{360} = \frac{10,2}{s} \cdot \frac{9}{360} = \frac{0,255}{s} \quad (\text{Ec.5})$$

$$\frac{\theta(s)}{U(s)} = \frac{Kp \cdot 0,255}{s + Kp \cdot 0,255} = \frac{1}{\frac{s}{Kp \cdot 0,255} + 1} \rightarrow \tau_{LC} = \frac{1}{Kp \cdot 0,255} \quad (\text{Ec.6})$$

La cantidad de muestreos ideal por cada τ es 5, teniendo en cuenta que el tiempo mínimo de muestreo está fijado en $T_s = 0,1 \text{ s}$ (limitación introducida al trabajar con Simulink en modo *external*):

$$\tau_{LC} = 5 \cdot T_s = 0,5 \quad (\text{Ec.7})$$

$$\tau_{LC} = \frac{1}{Kp \cdot 0,255} \rightarrow Kp = \frac{1}{0,5 \cdot 0,255} = 7,84 \quad (\text{Ec.8})$$

Cabe destacar que 7,84 es el valor calculado en condiciones ideales, pero dada la utilidad del robot de este proyecto, la cual no requiere precisión extrema, este valor puede ser aumentado ligeramente. Reduciendo la cantidad de muestreos por cada τ .

El esquema del lazo cerrado ideal tras los cálculos queda de la siguiente manera:

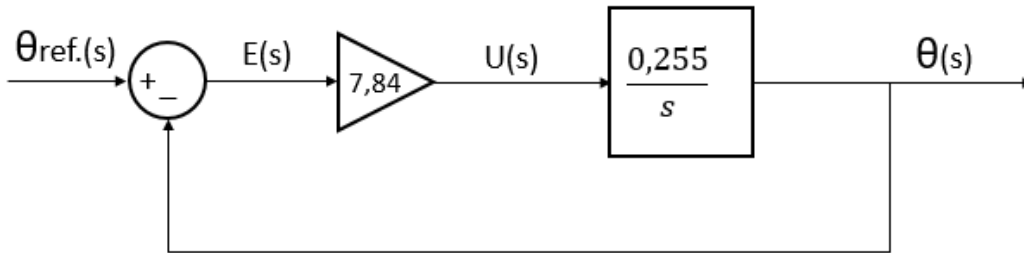


Ilustración 24: Lazo cerrado final motores grandes.

13.3.2. Motor mediano

De manera análoga al motor grande se ha calculado la planta y controlador de tipo P del motor mediano.

Teniendo como dato de partida que la velocidad angular máxima del motor mediano es de 250 rpm, la función de transferencia del motor puede aproximarse por:

$$250 \text{ rpm} \rightarrow 250 \cdot \frac{360^\circ}{60 \text{ s}} = 15 \text{ } \theta/\text{s} \quad (\text{Ec.9})$$

$$\frac{\Omega(s)}{U(s)} = K = 15 \quad (\text{Ec.10})$$

$$\frac{\theta}{U(s)} = \frac{K}{s} \cdot \frac{12}{36} = \frac{15}{s} \cdot \frac{12}{36} = 5 \quad (\text{Ec.11})$$

$$\frac{\theta(s)}{U(s)} = \frac{Kp \cdot 5}{s + Kp \cdot 5} = \frac{1}{\frac{s}{Kp \cdot 5} + 1} \rightarrow \tau_{LC} = \frac{1}{Kp \cdot 5} \quad (\text{Ec.12})$$

La cantidad de muestreos ideal por cada τ es 5, teniendo en cuenta que el tiempo mínimo de muestreo está fijado en $T_s = 0,1 \text{ s}$:

$$\tau_{LC} = 5 \cdot T_s = 0,5 \quad (\text{Ec.13})$$

$$\tau_{LC} = \frac{1}{Kp \cdot 5} \rightarrow Kp = \frac{1}{0,5 \cdot 5} = 0,4 \quad (\text{Ec.14})$$

El esquema del lazo cerrado del motor mediano es por tanto:

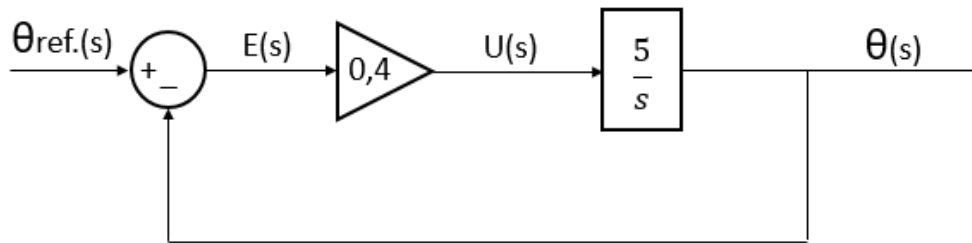


Ilustración 25: Lazo cerrado motor mediano.

13.4. Cálculo de ángulos a partir del punto

El bloque mostrado en la Ilustración 26 engloba los lazos de control de los tres motores explicados en el apartado anterior y además incluye las funciones definidas en código Matlab para obtener los ángulos de los dos engranajes en función del punto deseado al que se quiere llevar el rotulador.

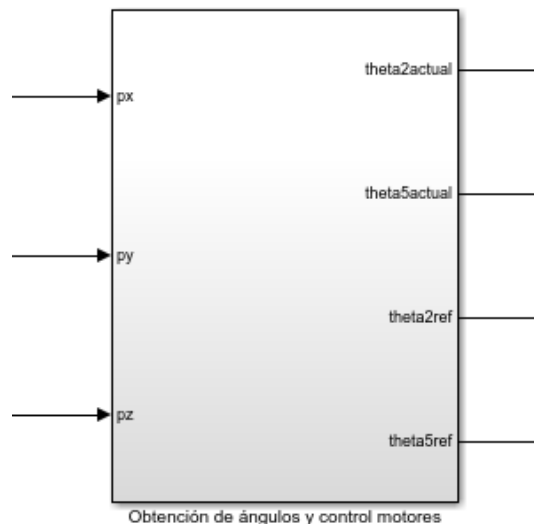


Ilustración 26: Bloque "cálculo de ángulos a partir del punto".

El bloque tiene como entradas las coordenadas X,Y del punto, utilizadas para obtener los ángulos como se explica a continuación y también la coordenada Z utilizada para el control del balancín.

Por otro lado, las salidas son el ángulo θ_2 y θ_5 de referencia, es decir, ángulo exacto que deben tomar los brazos del robot, así como los ángulos actuales en los que se encuentran los brazos. Estas cuatro salidas serán usadas posteriormente por un comparador.

A continuación, se desarrollan los cálculos realizados para la obtención de θ_2 y θ_5 en función de las coordenadas X e Y del punto P (P_y , P_x).

13.4.1. Cálculo del ángulo θ_2

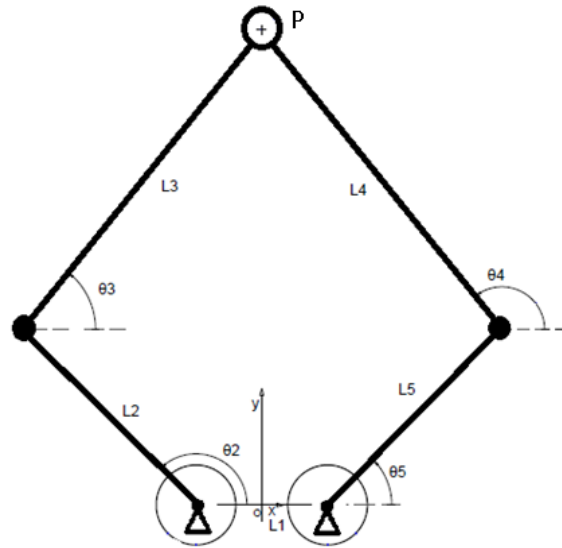


Ilustración 27: Mecanismo del robot.

Las expresiones para las coordenadas del punto P en función de los eslabones L2 y L3 son:

$$P_x = -\frac{L_1}{2} + L_2 \cos\theta_2 + L_3 \cos\theta_3 \quad (\text{Ec.15})$$

$$P_y = L_2 \sin\theta_2 + L_3 \sin\theta_3 \quad (\text{Ec.16})$$

Se despejan los términos en función de θ_3

$$P_x + \frac{L_1}{2} - L_2 \cos\theta_2 = L_3 \cos\theta_3 \quad (\text{Ec.17})$$

$$P_y - L_2 \sin\theta_2 = L_3 \sin\theta_3 \quad (\text{Ec.18})$$

Se elevan ambos lados de la ecuación al cuadrado

$$(P_x + \frac{L_1}{2} - L_2 \cos\theta_2)^2 = L_3^2 \cos^2 \theta_3 \quad (\text{Ec.19})$$

$$(P_y - L_2 \sin\theta_2)^2 = L_3^2 \sin^2 \theta_3 \quad (\text{Ec.20})$$

Se suman las dos ecuaciones y aplicando la identidad trigonométrica $\sin^2 \theta + \cos^2 \theta = 1$ el ángulo θ_3 desaparece de la ecuación, dejando solo L_3^2 , el cual es un dato conocido

$$(P_x + \frac{L_1}{2} - L_2 \cos\theta_2)^2 + (P_y - L_2 \sin\theta_2)^2 = L_3^2 \quad (\text{Ec.21})$$

A continuación, se desarrollan los cuadrados de los términos y se simplifica la función:

$$P_x^2 + \frac{L_1^2}{4} + L_2^2 \cos^2 \theta_2 + 2P_x \frac{L_1}{2} - 2P_x L_2 \cos \theta_2 - 2 \frac{L_1}{2} L_2 \cos \theta_2 + P_y^2 + L_2^2 \sin^2 \theta_2 - 2 P_y L_2 \sin \theta_2 = L_3^2 \quad (\text{Ec.22})$$

$$(-L_1 L_2 - 2P_x L_2) \cos \theta_2 + (-2 P_y L_2) \sin \theta_2 = L_3^2 - L_2^2 - \frac{L_1^2}{4} - P_x^2 - P_y^2 - P_x L_1 \quad (\text{Ec.23})$$

Se sustituyen los términos por las variables A2, B2 y C2 para hacer más sencillos el resto de cálculos.

$$A_2 \cos \theta_2 + B_2 \sin \theta_2 = C_2 \quad (\text{Ec.24})$$

La expresión genérica, dejando a un lado los subíndices es la siguiente.

$$A \cos \theta + B \sin \theta = C \quad (\text{Ec.25})$$

Se divide por $\cos \theta$ en ambos lados de la ecuación:

$$A + B \operatorname{tg} \theta = \frac{C}{\cos \theta} \quad (\text{Ec.26})$$

A continuación, se iguala la variable t a la tangente del ángulo mitad y se obtienen las expresiones del $\cos \theta$ y la $\operatorname{tg} \theta$ en función de la variable t para poder sustituirlas en la ecuación anterior.

$$t = \operatorname{tg} \frac{\theta}{2} = \sqrt{\frac{1 - \cos \theta}{1 + \cos \theta}} \quad (\text{Ec.27})$$

$$t^2 = \frac{1 - \cos \theta}{1 + \cos \theta} \quad (\text{Ec.28})$$

$$t^2 + t^2 \cos \theta = 1 - \cos \theta \quad (\text{Ec.29})$$

$$\cos \theta (t^2 + 1) = 1 - t^2 \quad (\text{Ec.30})$$

$$\cos \theta = \frac{1 - t^2}{1 + t^2} \quad (\text{Ec.31})$$

$$\cos 2\theta = \left(\frac{1 - t^2}{1 + t^2} \right)^2 \quad (\text{Ec.32})$$

$$\frac{1}{\cos^2 \theta} = \left(\frac{1 + t^2}{1 - t^2} \right)^2 = 1 + \operatorname{tg}^2 2\theta \quad (\text{Ec.33})$$

$$\operatorname{tg} \theta = \sqrt{\frac{(1 + t^2)^2}{(1 - t^2)^2} - 1} = \sqrt{\frac{4t^2}{(1 - t^2)^2}} \quad (\text{Ec.34})$$

$$\operatorname{tg} \theta = \frac{2t}{1 - t^2} \quad (\text{Ec.35})$$

Se sustituyen los términos citados anteriormente y se despeja la variable $t = \operatorname{tg} \frac{\theta}{2}$

$$A + B \frac{2t}{1-t^2} = \frac{1+t^2}{1-t^2} C \quad (\text{Ec.36})$$

$$(1-t^2) A + 2t B = (1+t^2) C \quad (\text{Ec.37})$$

$$A - t^2 A + 2t B - C - t^2 C = 0 \quad (\text{Ec.38})$$

$$(-A - C) t^2 + 2B t + (A - C) = 0 \quad (\text{Ec.39})$$

$$t = \frac{-2B \pm \sqrt{4B^2 + 4(A+C)(A-C)}}{-2(A+C)} \quad (\text{Ec.40})$$

$$\operatorname{tg} \frac{\theta}{2} = \frac{-B \pm \sqrt{B^2 + A^2 - C^2}}{-(A+C)} \quad (\text{Ec.41})$$

Por último, se despeja el ángulo θ , obteniendo la expresión final:

$$\theta = 2 \operatorname{tg}^{-1} \left(\frac{-B \pm \sqrt{B^2 + A^2 - C^2}}{-(A+C)} \right) \quad (\text{Ec.42})$$

La expresión para θ_2 , es por tanto:

$$\theta_2 = 2 \operatorname{tg}^{-1} \left(\frac{-B_2 \pm \sqrt{B_2^2 + A_2^2 - C_2^2}}{-(A_2 + C_2)} \right) \quad (\text{Ec.43})$$

Se debe tener cuidado ya que la expresión anterior tiene cuatro soluciones posibles, cada una de ellas se corresponde con las cuatro maneras de ensamblar los brazos para alcanzar el punto objetivo, tal como se observa en la Ilustración 28.

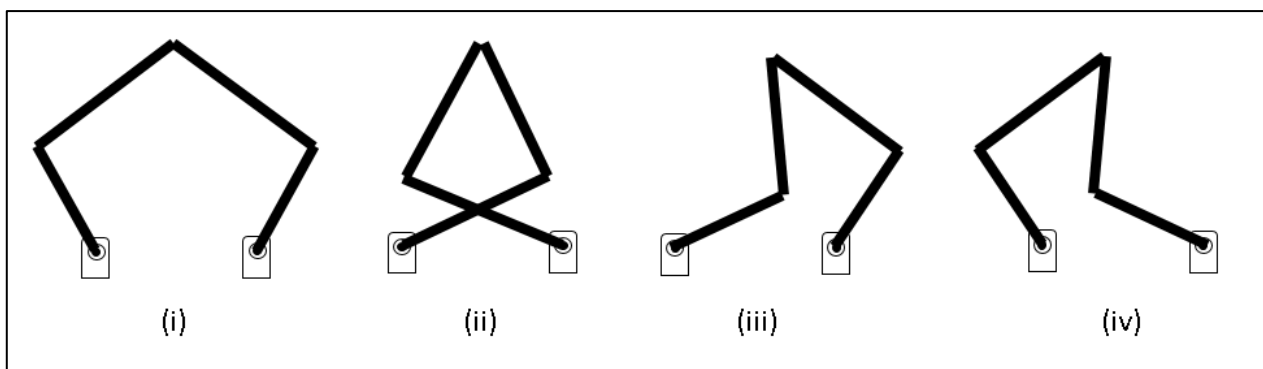


Ilustración 28: Posibles ensamblajes del mecanismo

Dado que el ensamblaje deseado es el (i) de la imagen anterior la expresión correspondiente para θ_2 es:

$$\theta_2 = 2 \operatorname{tg}^{-1} \left(\frac{-B_2 \pm \sqrt{B_2^2 + A_2^2 - C_2^2}}{-(A_2 + C_2)} \right) \quad (\text{Ec.44})$$

Siendo,

$$A_2 = -L_1 L_2 - 2P_x L_2 \quad (\text{Ec.45})$$

$$B_2 = -2 P_y L_2 \quad (\text{Ec.46})$$

$$C_2 = L_3^2 - L_2^2 - \frac{L_1^2}{4} - P_x^2 - P_y^2 - P_x L_1 \quad (\text{Ec.47})$$

13.4.2. Cálculo de θ_5

El cálculo de θ_5 se hace de manera análoga al de θ_2 , las expresiones de las coordenadas de P, esta vez en función de los eslabones L_5 y L_4 , son:

$$P_x = \frac{L_1}{2} + L_5 \cos \theta_5 + L_4 \cos \theta_4 \quad (\text{Ec.48})$$

$$P_y = L_5 \sin \theta_5 + L_4 \sin \theta_4 \quad (\text{Ec.49})$$

Se sigue el desarrollo de las expresiones:

$$P_x - \frac{L_1}{2} - L_5 \cos \theta_5 = L_4 \cos \theta_4 \quad (\text{Ec.50})$$

$$P_y - L_5 \sin \theta_5 = L_4 \sin \theta_4 \quad (\text{Ec.51})$$

$$\left(P_x - \frac{L_1}{2} - L_5 \cos \theta_5 \right)^2 = L_4^2 \cos^2 \theta_4 \quad (\text{Ec.52})$$

$$\left(P_y - L_5 \sin \theta_5 \right)^2 = L_4^2 \sin^2 \theta_4 \quad (\text{Ec.53})$$

$$\left(P_x - \frac{L_1}{2} - L_5 \cos \theta_5 \right)^2 + \left(P_y - L_5 \sin \theta_5 \right)^2 = L_4^2 \quad (\text{Ec.54})$$

$$P_x^2 + \frac{L_1^2}{4} + L_5^2 \cos^2 \theta_5 - 2P_x \frac{L_1}{2} - 2P_x L_5 \cos \theta_5 + 2 \frac{L_1}{2} L_5 \cos \theta_5 + P_y^2 + L_5^2 \sin^2 \theta_5 - 2 P_y L_5 \sin \theta_5 = L_4^2 \quad (\text{Ec.55})$$

$$(L_1 L_5 - 2P_x L_5) \cos \theta_5 + (-2 P_y L_5) \sin \theta_5 = L_4^2 - L_5^2 - \frac{L_1^2}{4} - P_x^2 - P_y^2 + P_x L_1$$

(Ec.56)

$$A_5 \cos \theta_5 + B_5 \sin \theta_5 = C_5 \quad (\text{Ec.57})$$

Obteniendo,

$$\theta_5 = 2 \operatorname{tg}^{-1} \left(\frac{-B_5 \pm \sqrt{B_5^2 + A_5^2 - C_5^2}}{-(A_5 + C_5)} \right) \quad (\text{Ec.58})$$

Teniendo en cuenta el ensamblaje, la expresión final para θ_5 es:

$$\theta_5 = 2 \operatorname{tg}^{-1} \left(\frac{-B_5 \pm \sqrt{B_5^2 + A_5^2 - C_5^2}}{-(A_5 + C_5)} \right) \quad (\text{Ec.60})$$

Siendo,

$$A_5 = L_1 L_5 - 2P_x L_5 \quad (\text{Ec.61})$$

$$B_5 = 2 P_y L_5 \quad (\text{Ec.62})$$

$$C_5 = L_4^2 - L_5^2 - \frac{L_1^2}{4} - P_x^2 - P_y^2 + P_x L_1 \quad (\text{Ec.63})$$

13.5. Error de aproximación al punto

Cuando el robot se mueve siguiendo una trayectoria se ha de controlar cuándo ha llegado al punto deseado, para poder mandarle las coordenadas del siguiente punto.

Para ello simplemente se ha colocado un bloque comparador a modo de realimentación que compara los ángulos θ_2 y θ_5 que actualmente presenta el mecanismo con los ángulos de referencia indicados. Si el valor absoluto de la diferencia entre ambos ángulos es inferior a $0,5^\circ$ la señal de salida llamada error tomará el valor de 0, en cualquier otro caso la señal de error tendrá el valor de 1. (anexo 2.4.)

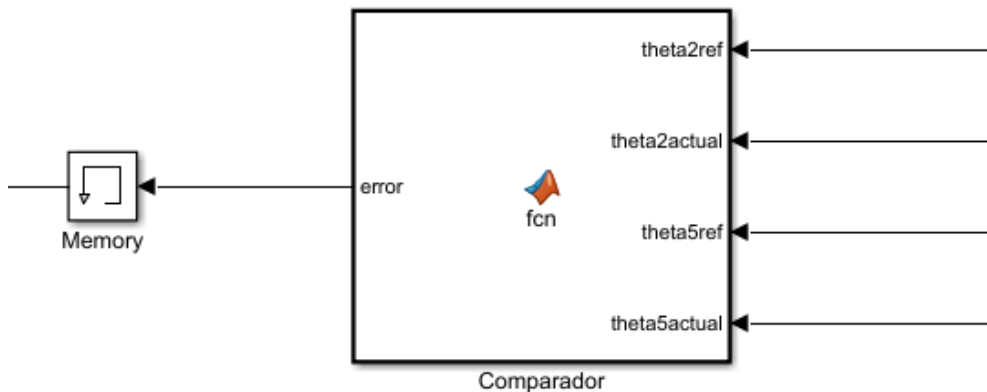


Ilustración 29: Bloque "Comparador".

Al tratarse la señal error de una señal de realimentación, es necesario que cuente con un bloque memoria a su salida, sin ella, el resto de bloques que cuentan con la señal realimentada como entrada no pueden realizar los cálculos debido a se producen bucles algebraicos en los que no se dispone de un valor inicial para realizar los cálculos.

El bloque memoria retrasa una muestra de tiempo el valor de la señal.

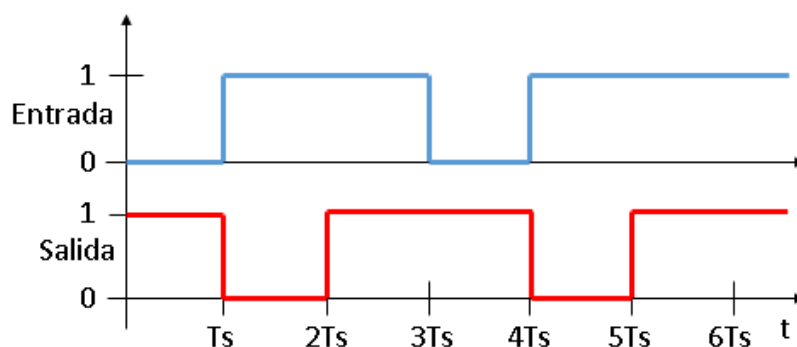


Ilustración 30: Comportamiento de la memoria.

Por defecto, cuando se inicia el programa, el robot no se encuentra en el punto deseado ya que primeramente resetea su posición tocando los sensores táctiles, es por ello que el valor inicial de la memoria ha de ser 1, indicando la existencia de un error de posición.

13.6. Obtención de coordenadas

El bloque obtención de coordenadas es el corazón del programa, ya que es el encargado de enviar punto a punto la posición a la que ha de moverse el robot, así como si tiene que hacerlo con el equipo de escritura levantado o bajado. Es decir, controla los 3 ejes del proyecto (X, Y, Z).

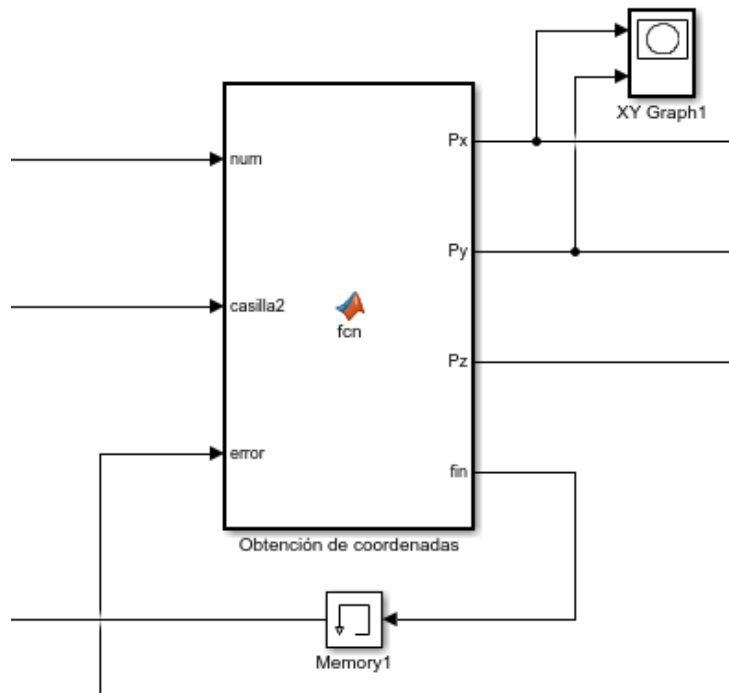


Ilustración 31: Bloque "Obtención de coordenadas".

Este bloque es alimentado con las siguientes entradas:

- num: número del 1 al 9 que se ha de escribir
- casilla2: casilla en la que se ha de escribir dicho número, este dato vendrá dado en forma de matriz de dimensiones 1x2. Siendo [1 1] la casilla superior izquierda del *sudoku* y [9 9] la casilla inferior derecha.
- error: Esta señal llega como realimentación del bloque explicado anteriormente

Con estos datos de entrada el bloque controla las coordenadas del punto en el que se encuentra el rotulador (Px, Py, Pz), y también tiene una salida llamada "fin" que tendrá un valor de 1 en el momento que el robot haya de terminado de escribir un número y 0 en cualquier otro momento.

Esta señal de fin es pasada a través de una memoria y enviada posteriormente al bloque explicado en el apartado 13.6. a modo de realimentación.

A continuación, se explica el bloque con mayor profundidad haciendo hincapié en los diferentes parámetros.

13.6.1. Número

Una vez se conoce el número (del 1 al 9) que el robot ha de escribir se mantiene dicho valor hasta que se modifica el valor de la casilla, como se explica en el próximo apartado. En ese momento se asigna a la variable “trayectoria” la trayectoria de los números, la cual, vendrá dada en forma de matriz de nx2. Habrá tantas filas (n) como puntos de la trayectoria se quieran definir, y las 2 columnas vienen dadas por las coordenadas x e y de cada punto (para ver las trayectorias de todos los números acudir al anexo 3).

13.6.2. Casilla

El proceso de escritura arranca cuando a este bloque le llega una casilla nueva, distinta a la anterior. Esto quiere decir que el proceso en la casilla actual, ha terminado y el robot debe desplazarse a la nueva. Para ello se comprueban las dos coordenadas de la casilla en muestras de tiempo consecutivas.

```
if(((casilla(1)-casilla_ant(1))~=0)||((casilla(2)-casilla_ant(2))~=0))
    casilla_nueva=1;
    casilla_ant(1)=casilla(1);
    casilla_ant(2)=casilla(2);
else
    casilla_nueva=0;
end
```

Al detectar una nueva casilla, la variable “casilla_nueva” toma el valor de 1 y este hecho desencadena el resto de sucesos:

- El contador “i” se pone a 1 (valor inicial).
- Se calculan las coordenadas del origen de la casilla con las siguientes operaciones:

```
origen=[0 0];
origen(1)= -50 + (10*casilla(2)); % -45 a 45 = 90mm ancho
origen(2) = 201.5-(10*casilla(1)); % 110 a 200 = 90mm alto
```

La coordenada X varía entre -40 y 40 en intervalos de 10. Coincidiendo con el centro de cada una de las 9 casillas, teniendo en cuenta que el *sudoku* mide 90mm de ancho y por tanto cada casilla 10mm.

Por otro lado la coordenada Y varía entre 111,5 y 191,5 en intervalos de 10. Esto se debe a que el número mide 7mm de alto, teniendo 1,5mm de margen tanto en la parte superior como inferior de la casilla.

- Se asigna a la variable “trayectoria” la matriz de la trayectoria del número a escribir.

- Se suma a la trayectoria el origen de la casilla, de tal manera que toda la trayectoria del número, que viene referenciada al origen (0,0) ahora pasa a estar ubicada en la casilla correspondiente:

```
trayectoria(:,1)=trayectoria(:,1)+origen(1);  
trayectoria(:,2)=trayectoria(:,2)+origen(2);
```

- Se obtiene el número de filas de la matriz trayectoria y por tanto el número de puntos a representar, es almacenado en la variable “nf”:

```
[nf,~]=size(trayectoria);
```

Todos estos pasos preparan y dejan actualizadas las variables que dominan el movimiento.

13.6.3. Coordenadas Px e Py

En este punto se comienza a asignar sucesivamente todos y cada uno de los puntos de las trayectorias de los números a las coordenadas X e Y del punto P (Px,Py) con la ayuda del contador “i”.

```
Px = trayectoria(i,1);  
Py = trayectoria(i,2);
```

Para aumentar dicho contador se han de cumplir simultáneamente las siguientes condiciones:

- error = 0 : La señal error es igual a cero cuando se ha alcanzado el punto P deseado.
- casilla_nueva = 0 : Es decir, que estemos todavía en la misma casilla y por tanto escribiendo el mismo número.
- i < nf : El contador “i” nunca puede sobrepasar ni igualar el número total de filas.

```
if ( (error==0) && (casilla_nueva==0) && (i<nf) )  
    i=i+1;  
end
```

13.6.4. Balancín, coordenada Pz

Una vez se tienen controladas las coordenadas X e Y se ha de controlar la coordenada Z del punto P, ya que este proyecto se desarrolla en el espacio.

La coordenada Z es gobernada por el balancín, cuando éste no ejerce fuerza el conjunto de escritura cae por su propio peso como se ha explicado anteriormente.

Mientras el contador “i” indique que el número está siendo escrito ($1 < i < nf$) el balancín estará inactivo (Balancín = 0) y por tanto dejará caer el rotulador para la escritura del número. En el momento que el contador sale de ese rango indica que el número ya ha sido escrito por lo que el balancín se activa (Balancín = 1) y separa el rotulador del papel.

```
if (i>1) &&(i<nf)
    Balancin=0;
else
    Balancin=1;
    if (i>=nf)
        fin=1;
    end
end

-Pz = Balancin;
```

Quedan así completamente definidas y controladas las tres coordenadas del punto $P = (Px, Py, Pz)$.

Por último, la señal “fin”, que indica el final de la escritura de un número, para así demandar el siguiente número a escribir y su casilla al bloque correspondiente, toma el valor 1 cuando el contador “i” alcanza o sobrepasa el número total de puntos que definen el número, es decir, todos han sido representados. Se recuerda que esta señal “fin” vuelve a tomar el valor 0 cuando una nueva casilla llega al bloque.

13.6.5. Visualización gráfica

Cuando el programa es ejecutado se visualiza en la gráfica XY el recorrido realizado en tiempo real por el robot. Se observan los números dibujados, así como la trayectoria recorrida para desplazarse de una casilla a otra. En la Ilustración 32 se ve el gráfico final tras la resolución completa de un *sudoku*.

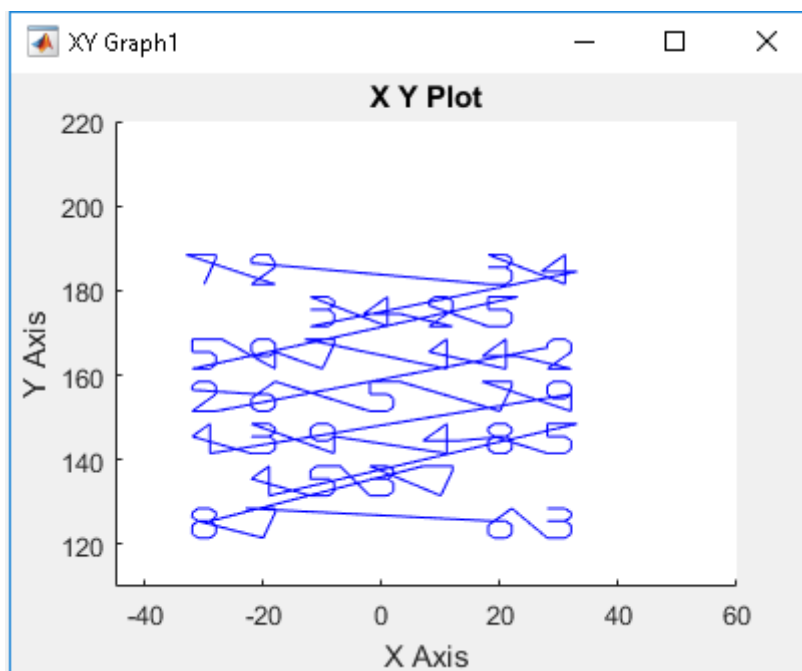


Ilustración 32: Visualización gráfica de la solución.

13.7. Obtención del número y casilla a partir del sudoku solución

Una vez se ha definido todo lo anterior solo cabe saber qué número y en qué casilla el robot debe escribir. Para ello se implementa el bloque “Datos a escribir”.

Este bloque tiene como entradas:

- huecos: matriz constante de 9x9, con 1 en aquellos lugares en los que hay un hueco y por tanto hay que escribir y 0 en aquellos ya rellenos en los que, por tanto, no hay que escribir. Se obtiene del workspace tras la resolución del *sudoku*.
- SUDOKU: matriz constante de 9x9, con todos los números que componen el *sudoku*, tanto los datos de partida como los resueltos por el programa. Se obtiene del workspace tras la resolución del *sudoku*.
- fin: señal que llega a modo de realimentación para indicar cuando ha terminado el programa de escribir el número anterior y por tanto se puede mandar el siguiente dato a escribir.

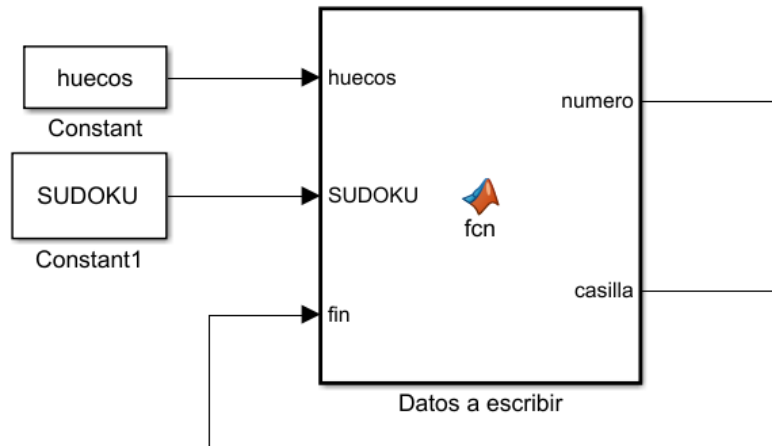


Ilustración 33: Bloque "datos a escribir".

El bloque recorre la matriz “huecos” por filas desde la casilla [1 1] hasta la [9 9] buscando dónde hay un hueco y por tanto se ha de escribir. Para ello se utiliza un contador llamado “c” que incrementa desde 1 a 81 (número total de casillas).

Para calcular la fila se realiza la siguiente operación:

$$F = \text{floor}((c - 1) / 9) + 1;$$

Donde la función floor() obtiene la parte entera de la división.

Y para calcular la columna se realiza la siguiente operación:

$$C = \text{mod}(c - 1, 9) + 1;$$

Donde la función mod() obtiene el resto de la división.

En el momento que se encuentra un 1 en la matriz huecos, es decir, hay un hueco en el que escribir el dato solución, el bloque actualiza el valor de sus salidas, siendo la casilla las coordenadas compuestas por [F C], y el número se corresponde con el ubicado en esta misma casilla de la matriz SUDOKU.

14. COMUNICACIÓN

14.1. Puertos utilizados

El bloque de programación tiene dos grupos de cuatro puertos cada uno. Uno de los grupos cuenta con los puertos 1, 2, 3 y 4 mientras que los puertos del otro grupo son denominados A, B, C y D.

En este proyecto los puertos con nombre numérico son utilizados para los sensores, por lo tanto, se recoge información a través de ellos, y los puertos nombrados con letras son utilizados para los actuadores, por estos se envía información.

Además, también se utiliza el puerto USB, en cual se conecta un módulo de WiFi portátil y gracias al cual el bloque de programación establece conexión con la red deseada.

Tabla 3: Puertos de comunicación.

| Puerto | Elemento | Tipo |
|---------------|------------------------------|--------------|
| A | Motor grande (θ_5) | Actuador |
| B | Motor mediano | Actuador |
| C | Motor grande (θ_2) | Actuador |
| 1 | Sensor táctil (θ_5) | Sensor |
| 4 | Sensor táctil (θ_2) | Sensor |
| USB | Módulo WiFi | Comunicación |

14.2. Tipos de conexión

Para comunicar al bloque de programación el Software que determina el comportamiento del robot es necesario primero establecer una conexión entre el PC y el Bloque EV3. Este bloque de programación de LEGO Mindsorms Education EV3 puede establecer conexión con un ordenador de tres formas diferentes:

- **Conexión inalámbrica—Bluetooth:** Para poder establecer una conexión Bluetooth entre el Bloque EV3 y el equipo que ejecuta el Software de EV3, se debe activar la funcionalidad Bluetooth en el Bloque EV3.
- **Cable USB:** Con el cable USB, se conecta el extremo Mini USB al puerto para PC del Bloque EV3 (ubicado junto al Puerto D). Y el otro extremo USB al equipo.
- **Conexión inalámbrica—Wi-Fi:** Esta opción se explica en detalle en el apartado inmediatamente siguiente.

14.3. Procedimiento de conexión inalámbrica Wi-Fi

Esta ha sido la opción utilizada en el proyecto, ya que el cable USB limita físicamente la distancia entre el PC y el robot, resultando molesto en el montaje. La opción de conexión por Bluetooth se descartó debido a que el PC utilizado no cuenta con este tipo de conexión, no obstante, se puede establecer la conexión Bluetooth incorporando un conector USB con esta función al equipo.

Para establecer la conexión a través de Wi-Fi, el primer paso es adquirir un conector de WiFi tipo USB Netgear N150 para red inalámbrica, y conectarlo en el puerto USB del Bloque EV3.

Una vez se tiene el conector USB se procede al encendido del Bloque EV3, para ello se ha de mantener pulsado el botón Centro hasta que la luz de la botonera se ilumine en rojo. En ese momento se puede dejar de presionar el botón y se ha de esperar hasta que dicha luz cambie a color verde, indicando que el sistema se ha encendido completamente y está listo para su uso.

A continuación, usando los botones Derecha e Izquierda se accede a la pantalla de Ajustes del Bloque donde se debe seleccionar la palabra *“Wi-Fi”* utilizando los botones Arriba y Abajo y presionar el botón Centro para aceptar. Aparecerá una subpantalla en la que se debe volver a seleccionar la opción *“Wi-Fi”* para activar así este tipo de conexión. Si el proceso ha sido exitoso aparecerá una marca de verificación en la casilla Wi-Fi.

El Wi-Fi ahora está activado en el Bloque EV3 y aparecerá un icono de Wi-Fi en la esquina superior izquierda de la pantalla.

Para visualizar las redes Wi-Fi disponibles y poder elegir así la deseada, se debe volver a entrar a la opción de *“Wi-Fi”* en la pantalla de Ajustes, pero esta vez se elige la opción *“connections”*, el dispositivo inicia la búsqueda de redes y muestra en pantalla todas las opciones. Se selecciona con los botones Arriba y Abajo la red deseada y se acepta con el botón Centro. Una vez dentro la red se pulsa el botón Centro nuevamente sobre la opción *“connect”*.

Se despliega una subpantalla con el nombre *“Encryption”*. El Bloque EV3 solo puede establecer conexión con redes cuyos modos de cifrado sean: ninguno o WPA2. En caso de no tener ningún tipo de cifrado se elige la opción *“None”* y la conexión será inmediata, si por el contrario la red cuenta con un cifrado WPA2, se elige la opción *“WPA2”* y se despliega un teclado con el fin de introducir la contraseña correspondiente, en el que se navega con los botones Arriba, Abajo, Izquierda y Derecha y se selecciona el carácter con el botón Centro. Cabe destacar que dicho teclado cuenta exclusivamente con números y letras mayúsculas y minúsculas, a excepción de la ñ, por lo que si la contraseña cuenta con algún símbolo o carácter especial no podrá ser introducida. Si la contraseña es correcta se establece la conexión sin ningún problema.

Una vez el Bloque está conectado a una red Wi-Fi se hace lo mismo con el equipo PC. Es condición imprescindible que ambos dispositivos se conecten a la misma red para poder establecer la conexión entre ellos.

14.4. Parámetros en Simulink

Tras conectar el Bloque EV3 y el PC a la misma red Wi-Fi, en Simulink se accede a la pestaña “Tools”, “Run on Target Hardware”, “Options...” (Ilustración 34)

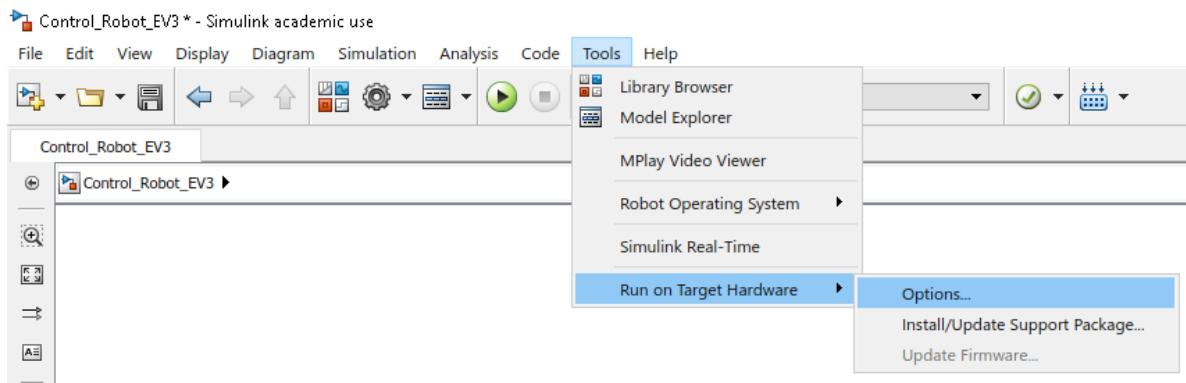


Ilustración 34: Conexión Simulink 1.

Se despliega la pantalla de la Fig.kjdshfjdgfjg. En la opción “Hardware board” se elige “LEGO MINDSTORMS EV3”, en “Groups” se selecciona “Host to Target Connection” y en “Connection type” la opción “WiFi”.

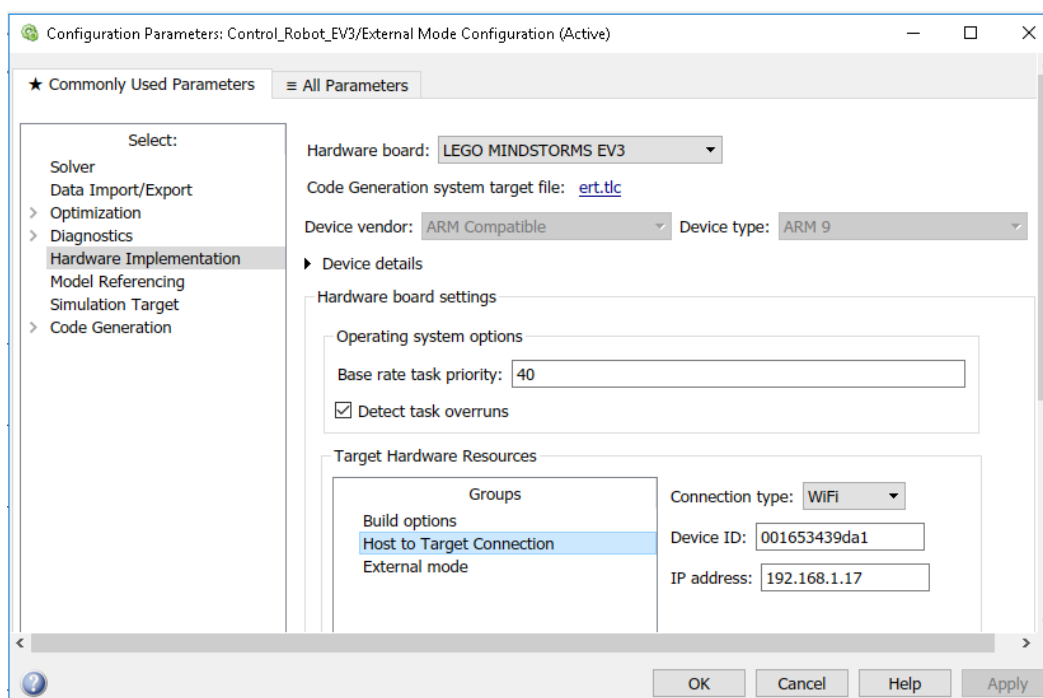


Ilustración 35: Conexión Simulink 2.

Para finalizar la configuración se ha de introducir la ID y la IP del Bloque EV3, en caso de desconocer estos datos se ha de ir a la pestaña de Ajustes del Bloque EV3, entrando en la opción “Brick info” con el botón Centro se despliega toda la información del Bloque. Se introduce la ID en “Device ID” y la dirección IP en “IP addresss” y por último pulsamos el botón OK para guardar esta configuración (Ilustración 35).

15. RESOLUCIÓN POR MEDIO DEL ROBOT

Tras haber sido explicados todos los pasos realizados en el proyecto a lo largo del documento, queda visualizar la resolución física del *sudoku* por parte del robot. Es decir, la escritura de los números en sus casillas correspondientes con la ayuda de un rotulador.

15.1 Referencia en el folio

Con la finalidad de que el robot se ubique siempre en el mismo lugar respecto a la cuadrícula del *sudoku*, el folio para la resolución (anexo 4) cuenta con una marca en forma de cruz que ha de coincidir con el origen del mecanismo de los brazos (punto medio del eslabón L1).

Se han introducido tres barras finas a través de los orificios que actúan como marcadores y facilitan la colocación del robot. (Ilustración 36).

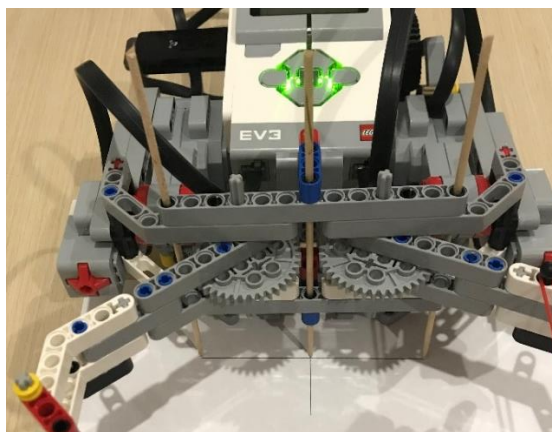


Ilustración 36: Referencia en el folio.

Una vez ubicado en el lugar correspondiente la posición del robot respecto al *sudoku* queda de la siguiente manera:

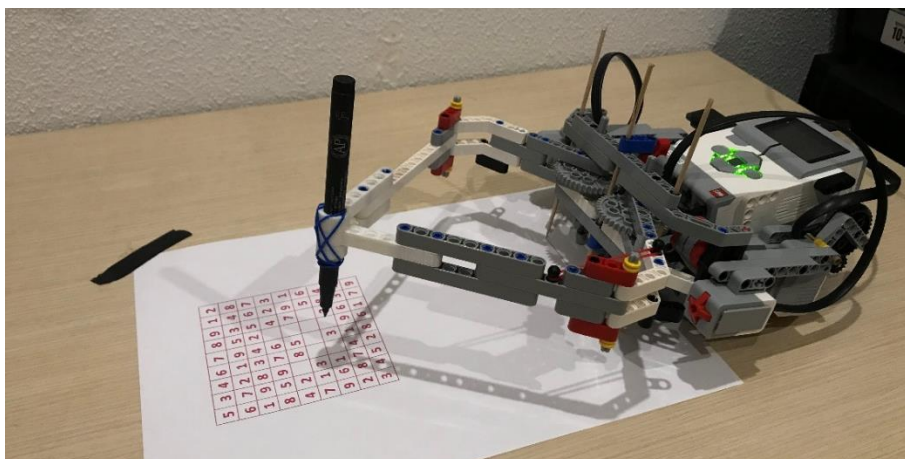


Ilustración 37: Ubicación del robot.

15.2 Solución

Como demostración de este trabajo se ha solucionado el *sudoku* de la Ilustración 38, el cual ha sido diseñado en tinta roja con el fin de contrastar con la solución, escrita con un rotulador de tinta negra.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | | 4 | 2 | 3 |
| 4 | 2 | | 8 | 5 | | 7 | 9 | 1 |
| 7 | 1 | 3 | | | | | 5 | 6 |
| 9 | 6 | 1 | | 3 | | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Ilustración 38: Sudoku a resolver.

Tras ejecutar el programa y permitir al robot rellenar las casillas correspondientes el resultado es el siguiente:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Ilustración 39: Sudoku resuelto por el robot.

Todos los números del 1 al 9 han sido escritos en esta solución, de manera que se puede apreciar el diseño de todos ellos. Los números han sido colocados en la casilla correspondiente según las normas de resolución, es decir, la solución es correcta y, a pesar de no ser tan perfectos como los escritos a ordenador, son completamente legibles.

16. CONCLUSIONES

Este trabajo ha sido desarrollado gradualmente por etapas, y cada una de ellas ha presentado diferentes niveles de dificultad.

Uno de los pasos dados en la etapa temprana de desarrollo del proyecto fue el diseño y montaje físico del robot. A priori fue un paso sencillo, pero más adelante, tras poner en funcionamiento el robot, se tuvieron que hacer modificaciones en el diseño, ya que, algunas de las piezas rozaban entre sí haciendo que el rotulador avanzase de forma tortuosa.

La programación en código Matlab tanto de la resolución del *sudoku*, como de los bloques de control del robot se desarrolló de forma fluida, a pesar de los múltiples errores que se han de subsanar siempre que se realiza un código de estas características. Surgieron problemas del estilo de bucles infinitos de los que el programa no conseguía salir por estar mal definidos, incompatibilidad de tipos de datos al realizar operaciones o variables que han de ser persistentes para que su valor sea recordado y actualizado cada vez que se accede a la función.

Por último, cabe destacar la gran cantidad de cambios en tamaños de número y casillas, así como en tolerancias de ángulos, que se han tenido que realizar y probar de manera experimental, hasta dar con la precisión requerida para que el número a dibujar encajase en la casilla correspondiente sin invadir otras casillas.

17. VÍAS DE CONTINUACIÓN

Este trabajo ha sido desarrollado por completo y cumple con todos los objetivos marcados inicialmente. Pero sí es cierto que está abierto a mejoras y ampliaciones, por ejemplo:

- Integrar un sensor óptico encargado de escanear los datos de partida del *sudoku*. Esta opción aumentaría la complejidad del código, así como el tiempo de ejecución, ya que es un proceso lento. Pero por otro lado el robot gana autonomía porque no requiere la acción del usuario para introducir dichos datos.
- Este proyecto podría ser desarrollado con un montaje más avanzado, utilizando componentes de un carácter más profesional y mejorando así sus prestaciones y precisión.

Las piezas utilizadas en este trabajo pertenecen a un kit de Lego que, a pesar de su gran potencial, no deja de ser un kit didáctico, compuesto de piezas plásticas.





**UNIVERSIDAD
DE LA RIOJA**

Documento 3:

ANEXOS

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

ÍNDICE DE ANEXOS

| | |
|---|----|
| 1. ESQUEMAS SIMULINK..... | 73 |
| 1.1. Esquema principal | 73 |
| 1.2. Esquema “Obtención de ángulos y control motores” | 74 |
| 1.3. Esquema “Motor C” | 75 |
| 1.4. Esquema “Motor A” | 75 |
| 1.5. Esquema “Motor B” | 75 |
| 2. CÓDIGO MATLAB | 76 |
| 2.1. Resolución del sudoku | 76 |
| 2.2. Función “Datos a escribir” | 78 |
| 2.3. Función “Obtención de coordenadas” | 79 |
| 2.4. Función “Comparador” | 83 |
| 2.5. Función “Cálculo de ángulos a partir de coordenadas” | 83 |
| 2.6. Función “RESET” del motor C | 84 |
| 2.7. Función “RESET” del motor A | 84 |
| 3. DISEÑO DE LA TRAYECTORIA DE LOS NÚMEROS | 85 |
| 4. PLANTILLA PARA LA RESOLUCIÓN..... | 90 |
| 5. INSTRUCCIONES DE MONTAJE..... | 92 |

1. ESQUEMAS SIMULINK

1.1. Esquema principal

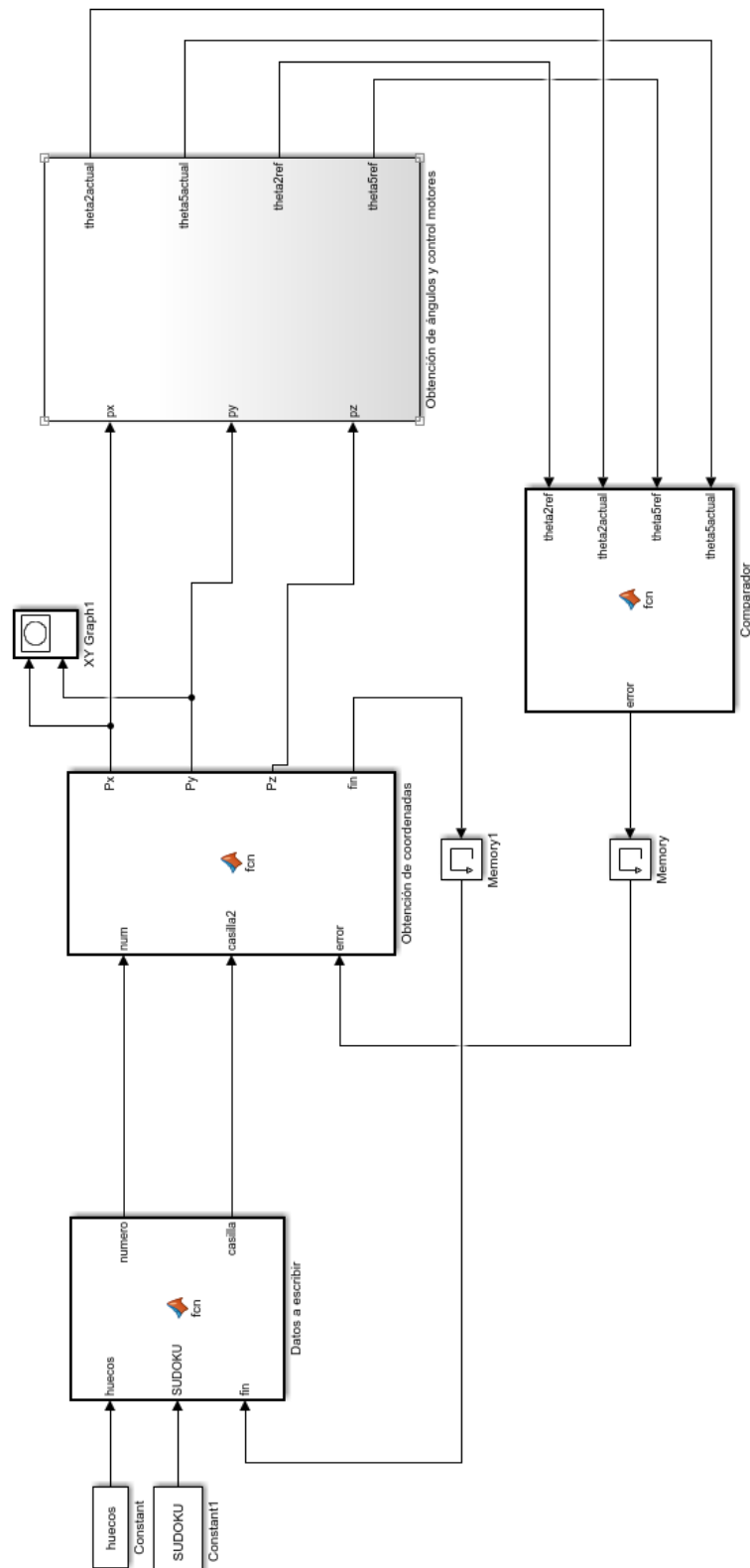


Ilustración 40: Esquema general Simulink.

1.2. Esquema “Obtención de ángulos y control motores”

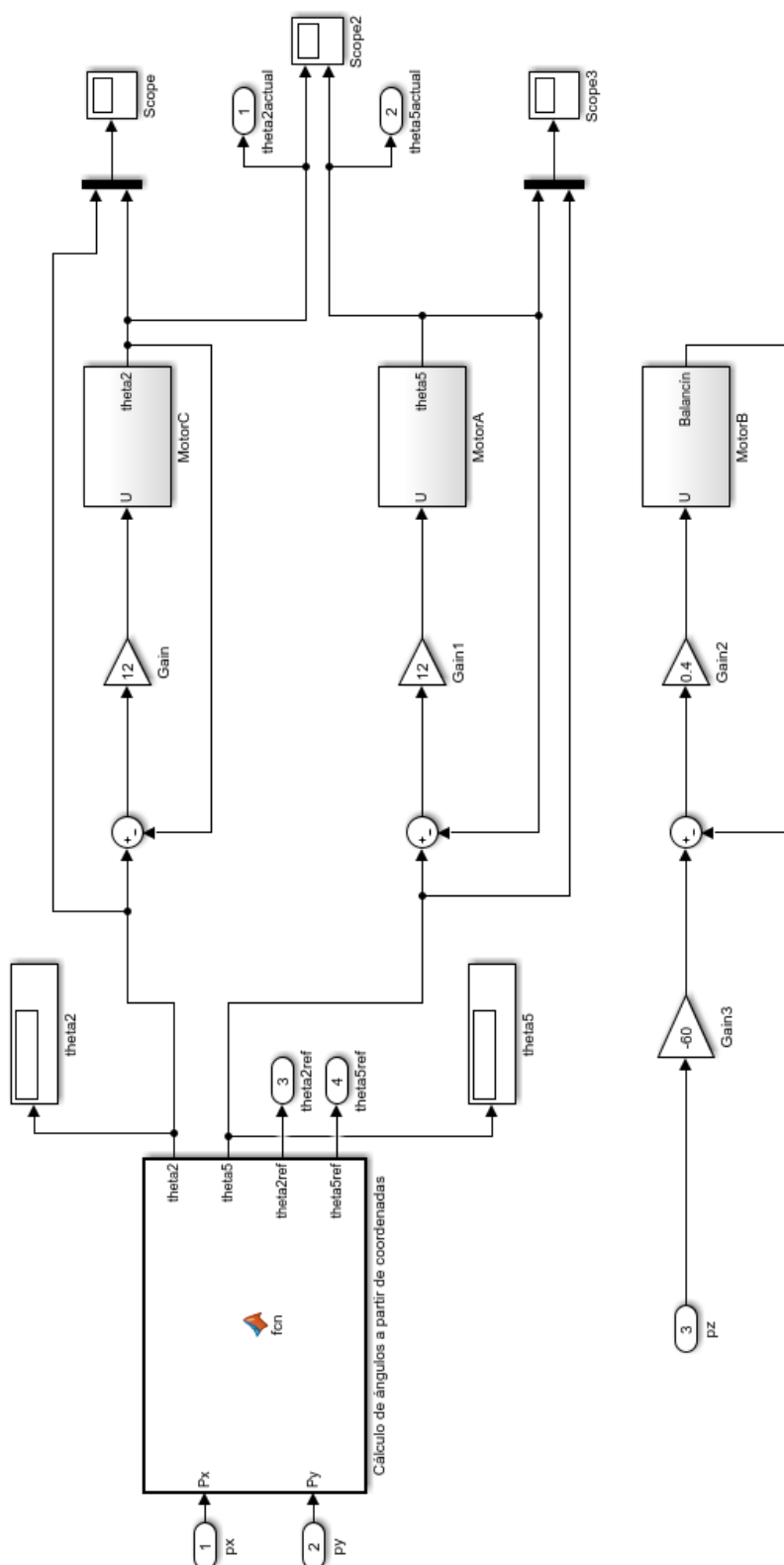


Ilustración 41: Esquema Simulink “Obtención de ángulos y control motores”

1.3. Esquema “Motor C”

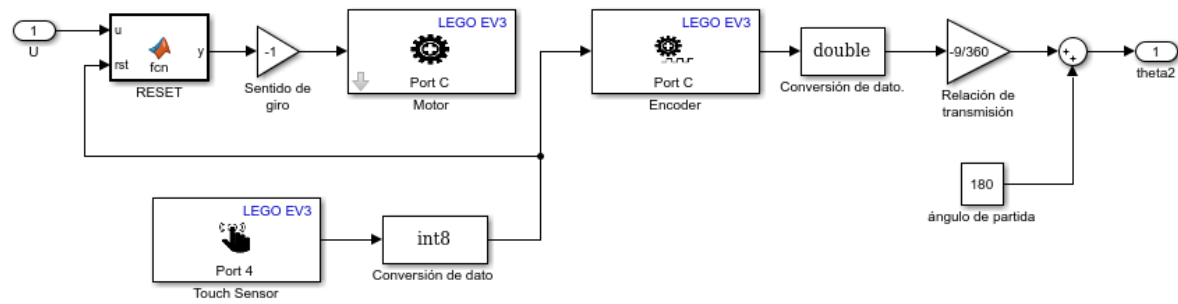


Ilustración 42: Esquema "Simulink motor C".

1.4. Esquema “Motor A”

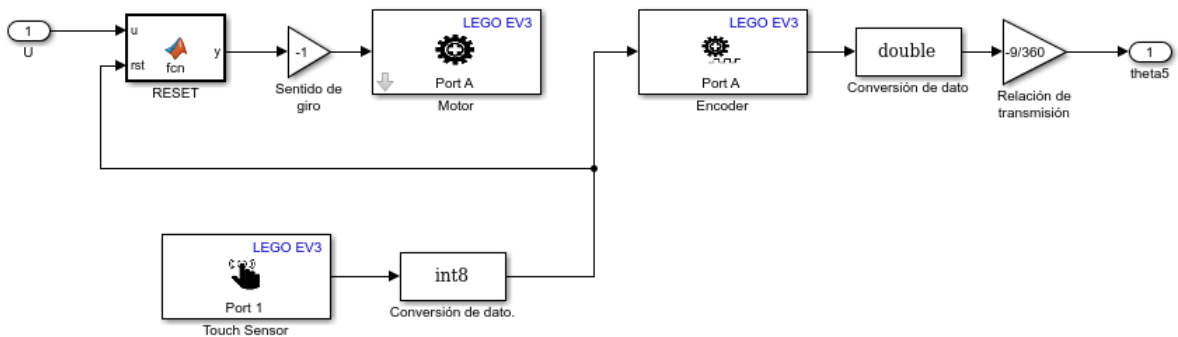


Ilustración 43: Esquema Simulink "motor A".

1.5. Esquema “Motor B”

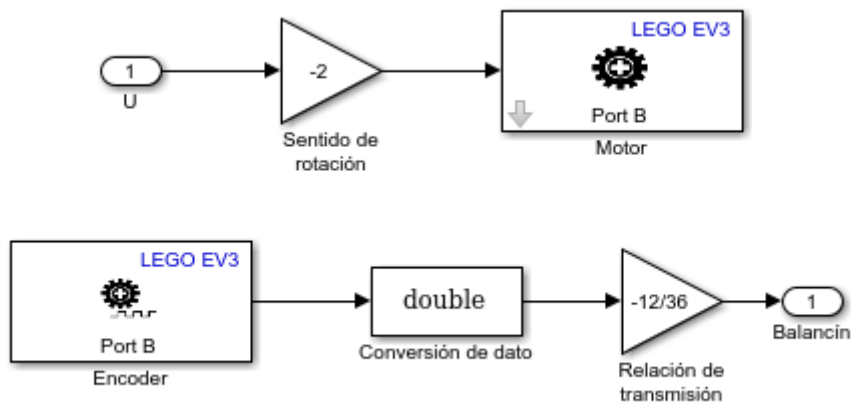


Ilustración 44: Esquema Simulink "motor B"

2. CÓDIGO MATLAB

2.1. Resolución del sudoku

```
% Se crea la matriz "huecos" de 9x9 la cual indicará dónde hay un
espacio
% y dónde un dato ya escrito
% Se llena de ceros
huecos=zeros(9,9);

% A continuación se revisan todas las columnas (C) y filas (F)
% En aquellos espacios sin datos se coloca un 1
for F=1:9
    for C=1:9
        if SUDOKU(F,C)==0
            huecos(F,C)=1;
        end
    end
end

F=1; % Se inicializa el n° de fila
valido=1; % Se inicializa la variable válido
while F<=9
    C=1; % Se inicializa el n° de columna
    while C<=9
        if huecos(F,C)==1 % Si en la posición hay un 1 = no hay dato
            for (a = SUDOKU(F,C)+1:9) % a toma valores entre 1 y 9
                SUDOKU(F,C)=a;
                valido=1; % Si válido

                %Comprobamos que el n°"a" no se encuentre ya en esa
fila
                for (m=1:9)
                    if (SUDOKU(F,C)==SUDOKU(F,m) && C~=m)
                        valido=0; % No válido
                    end

                    %Igualmente para la columna
                    if (SUDOKU(F,C) == SUDOKU(m,C) && F~=m)
                        valido=0; % No válido
                    end
                end

                %Se comprueba que el n°"a" no se encuentre ya en el
cuadrado
                C_x=floor((C-1)/3);
                C_y=floor((F-1)/3);
                for (p=1:3)
                    for (q=1:3)
                        if SUDOKU(F,C)==SUDOKU(3*C_y+p,3*C_x+q) &&
(F~=3*C_y+p || C~=3*C_x+q)
                            valido=0; % No válido
                        end
                    end
                end
            end
        end
    end
    F=F+1;
end
```

```
% Si el número es valido se guarda como solución
if (valido==1)
    break;
end
end
end

% Si el número no es válido
if (valido==0)
    SUDOKU(F,C)=0; % Se reestablece el valor de 0 en el Sudoku

    % Busca el anterior dato escrito. Y se incrementa su valor
    % en 1 unidad.
    if (C==1)
        for (r=F-1:-1:1)
            for (s=9:-1:1)
                if (huecos(r,s)==1)
                    F=r;
                    C=s-1;
                    % se resta 1 a la columna ya que es
                    incrementada
                    %posteriormente
                    break;
                end
            end
            if (huecos(r,s)==1)
                break;
            end
        end
    else
        flag=0;
        for (r=9:-1:1)
            for (s=9:-1:1)
                if ((r<F) || (r==F&&s<C)) && (huecos(r,s)==1)
                    F=r;
                    C=s-1;
                    flag=1;
                    break;
                end
            end
            if (flag==1)
                break;
            end
        end
    end
    end
    %Una vez encontrado el dato escrito anterior, se prueba
    con el
    %siguiente número al que se probó previamente. Si tras
    probar
    %con todos no se encuentra una opción válida es porque el
    % error viene de un número anterior, que será encontrado
    % tras ejecutar este algoritmo repetidas veces.
    %el error es debido a que en algunas casillas hay varias
    %opciones y hasta que no provocan un futuro error no se
    % puede saber cuál de las opciones era la correcta
    end
    C=C+1; % Incremento del n° de columna
    end
    F=F+1; % Incremento del n° de fila
    end
```

2.2. Función “Datos a escribir”

```
function [numero, casilla]= fcn(huecos, SUDOKU, fin)

% c= contador
% C= columna
% F= fila
% fin_ant= valor anterior de la variable fin

numero=0;
casilla=[0 0];

persistent c fin_ant C F
if isempty(c)
    c=0;
    fin_ant=0;
    C=1;
    F=1;
end

if (fin==1)&&(fin_ant==0) %flanco ascendente en la variable fin
    aux=0;
    while (aux==0) %mientras no se encuentre un hueco
        if (c<=81)
            c=c+1; %se incrementa el contador
        end
        C=mod(c-1,9)+1; %cálculo de la columna
        F=floor((c-1)/9)+1; %cálculo de la fila
        if (huecos(F,C)==1)
            aux=1;
        end
    end
end

numero= SUDOKU(F,C);
casilla(1)=F;
casilla(2)=C;
fin_ant=fin;
```

2.3. Función “Obtención de coordenadas”

```
function [Px, Py, Pz, fin] = fcn(num,casilla2,error)

casilla=[0 0];
casilla=casilla2;
fin=0;

%VARIABLES PERSISTENTES
persistent i casilla_ant casilla_nueva trayectoria nf;
if isempty(i)
    i=1;
    casilla_ant=[0 0];
    casilla_nueva=1;
    trayectoria=[0 0];
    nf=0;
end

if (((casilla(1)-casilla_ant(1))~=0) || ((casilla(2)-
casilla_ant(2))~=0))
    casilla_nueva=1;
    casilla_ant(1)=casilla(1);
    casilla_ant(2)=casilla(2);
else
    casilla_nueva=0;
end

if (casilla_nueva==1)

    i=1;

    % CÁLCULO DEL PUNTO DE ORIGEN (x,y) A PARTIR DE LA CASILLA
    origen=[0 0];
    origen(1)= -50 + (10*casilla(2)); % -45 a 45 = 90mm ancho
    origen(2) = 201.5-(10*casilla(1)); % 110 a 200 = 90mm alto

    % TRAYECTORIAS DE LOS NÚMEROS
    if num==1
        trayectoria=[1 0;
            1 0;
            1 1;
            1 2;
            1 3;
            1 4;
            1 5;
            1 6;
            1 7;
            0 6;
            -1 5;
            -2 4;
            -2 4;]
    elseif num==2
        trayectoria=[2 0;
            2 0;
            1 0;
            0 0;
            -1 0;
            -2 0;
```

```
-1 1;  
0 2;  
1 3;  
2 4;  
2 5;  
2 6;  
1 7;  
0 7;  
-1 7;  
-2 6;  
-2 5;  
-2 5;]  
elseif num==3  
trayectoria=[-2 0;  
-2 0;  
-1 0;  
0 0;  
1 0;  
2 1;  
2 2;  
2 3;  
1 4;  
0 4;  
-1 4;  
-2 4;  
-1 4;  
0 4;  
1 4;  
2 5;  
2 6;  
1 7;  
0 7;  
-1 7;  
-2 7;  
-2 7;]  
elseif num==4  
trayectoria=[1 0;  
1 0;  
1 1;  
1 2;  
1 3;  
1 4;  
1 5;  
1 6;  
1 7;  
0 6;  
-1 5;  
-2 4;  
-3 3;  
-2 3;  
-1 3;  
0 3;  
1 3;  
2 3;  
3 3;  
3 3;]  
elseif num==5  
trayectoria=[-2 0;  
-2 0;  
-1 0;  
0 0;
```

```
1 0;
2 1;
2 2;
2 3;
1 4;
0 4;
-1 4;
-2 4;
-2 5;
-2 6;
-2 7;
-1 7;
0 7;
1 7;
2 7;
2 7;]
elseif num==6
trayectoria=[-2 4;
-2 4;
-1 4;
0 4;
1 4;
2 3;
2 2;
2 1;
1 0;
0 0;
-1 0;
-2 1;
-2 2;
-2 3;
-2 4;
-2 5;
-2 6;
-1 7;
0 7;
1 7;
2 7;
2 7;]
elseif num==7
trayectoria=[0 0;
1 3;
2 6;
2 7;
1 7;
0 7;
-1 7;
-2 7;
-2 7;]
elseif num==8
trayectoria=[1 4;
1 4;
2 3;
2 2;
2 1;
1 0;
0 0;
-1 0;
-2 1;
-2 2;
-2 3;
```



```
-1 4;
0 4;
1 4;
2 5;
2 6;
1 7;
0 7;
-1 7;
-2 6;
-2 5;
-1 4;
-1 4;]
elseif num==9
trayectoria=[2 0;
2 0;
2 1;
2 2;
2 3;
2 4;
2 5;
2 6;
1 7;
0 7;
-1 7;
-2 6;
-2 5;
-2 4;
-1 3;
0 3;
1 3;
2 4;
2 4;]

end

trayectoria(:,1)=trayectoria(:,1)+origen(1);
trayectoria(:,2)=trayectoria(:,2)+origen(2);

[nf,~]=size(trayectoria);
end

%SALIDA DE DATOS
Px = trayectoria(i,1);
Py = trayectoria(i,2);

if ((error==0)&&(casilla_nueva==0)&&(i<nf))
    i=i+1;
end

%CONTROL DEL BALANCÍN
if (i>1)&&(i<nf)
    Balancin=0;
else
    Balancin=1;
    if (i>=nf)
        fin=1;
    end
end

Pz = Balancin;
```

2.4. Función “Comparador”

```
function error = fcn(theta2ref, theta2actual, theta5ref, theta5actual)

if (abs(theta2actual-theta2ref)<=0.5) && (abs(theta5actual-
theta5ref)<=0.5))
    error=0;
else
    error=1;
end
```

2.5. Función “Cálculo de ángulos a partir de coordenadas”

```
function [theta2, theta5, theta2ref, theta5ref] = fcn(Px, Py)

%Longitud de los eslabones en mm:
L1= 48;
L2= 90;
L3= 140;
L4= 140;
L5= 90;

%Agrupaciones de términos para simplificar las expresiones:

%Theta2
A2= -L1*L2-2*Px*L2;
B2= -2*Py*L2;
C2= (L3^2)-(L2^2)-((L1/2)^2)-(Px^2)-(Py^2)-Px*L1;

%Theta5
A5= L1*L5-2*Px*L5;
B5= -2*Py*L5;
C5= (L4^2)-(L5^2)-((L1/2)^2)-(Px^2)-(Py^2)+Px*L1;

%Ecuaciones:

theta2_mas= 2*atand((-B2 + sqrt((B2^2)+(A2^2)-(C2^2)))/(-(A2+C2)));

%theta2_menos= 2*atand((-B2 - sqrt((B2^2)+(A2^2)-(C2^2)))/-(A2+C2));

%theta5_mas= 2*atand((-B5 + sqrt((B5^2)+(A5^2)-(C5^2)))/-(A5+C5));

theta5_menos= 2*atand((-B5 - sqrt((B5^2)+(A5^2)-(C5^2)))/-(A5+C5));

%A continuación se acota el valor de los ángulos:
%En caso de que el valor del ángulo se salga del rango,
%se fija en el límite más cercano

if (theta2_mas<-45)
    theta2_mas=180;
end

if ((theta2_mas<90) && (theta2_mas>=-45))
    theta2_mas=90;
```

```
end

if ((theta5_menos>90) || (theta5_menos<-135))
    theta5_menos=90;
end

if ((theta5_menos<0) && (theta5_menos>=-135))
    theta5_menos=0;
end

theta2 = theta2_mas;
theta5 = theta5_menos;

theta2ref=theta2;
theta5ref=theta5;
```

2.6. Función “RESET” del motor C

```
function y = fcn(u,rst)

persistent rstOK;
if isempty(rstOK)
    rstOK=0;
end

if (rst==1)
    rstOK=1;
end

if (rstOK==1)
    y=u;
else
    y=50;
end
```

2.7. Función “RESET” del motor A

```
function y = fcn(u,rst)

persistent rstOK;
if isempty(rstOK)
    rstOK=0;
end

if (rst==1)
    rstOK=1;
end

if (rstOK==1)
    y=u;
else
    y=-50;
end
```

3. DISEÑO DE LA TRAYECTORIA DE LOS NÚMEROS

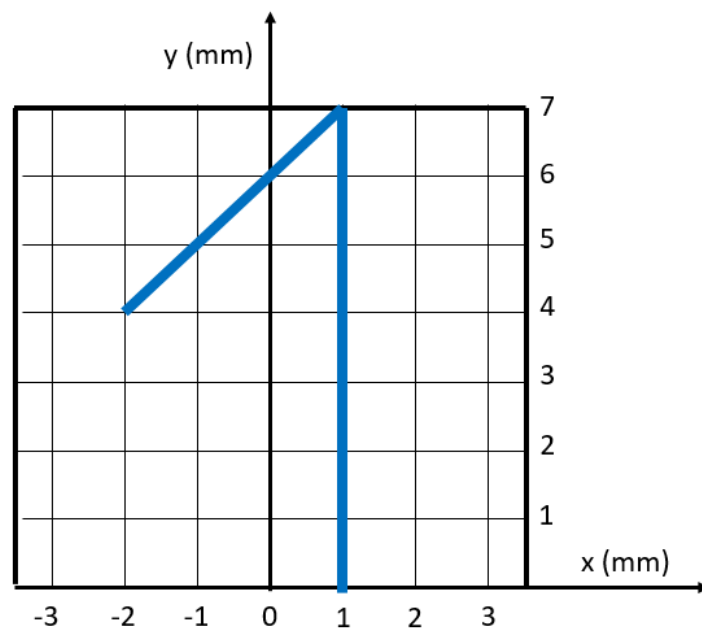


Ilustración 45: Diseño del número 1.

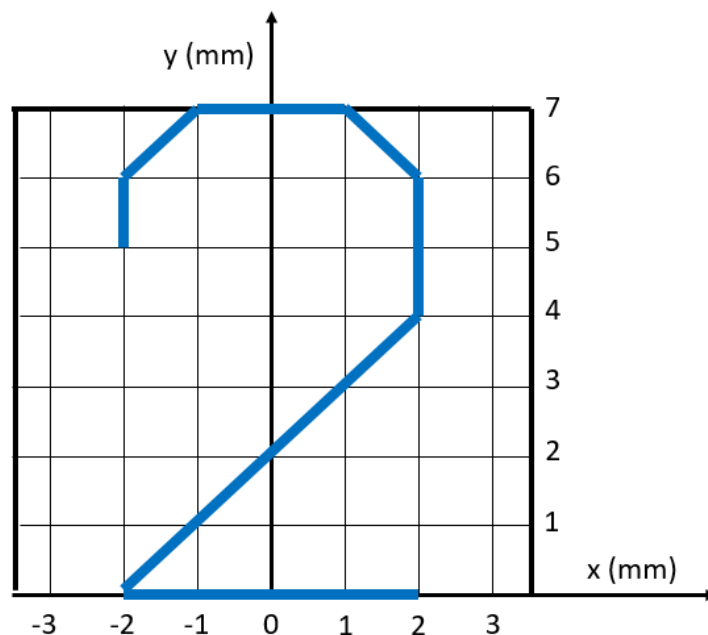


Ilustración 46: Diseño del número 2.

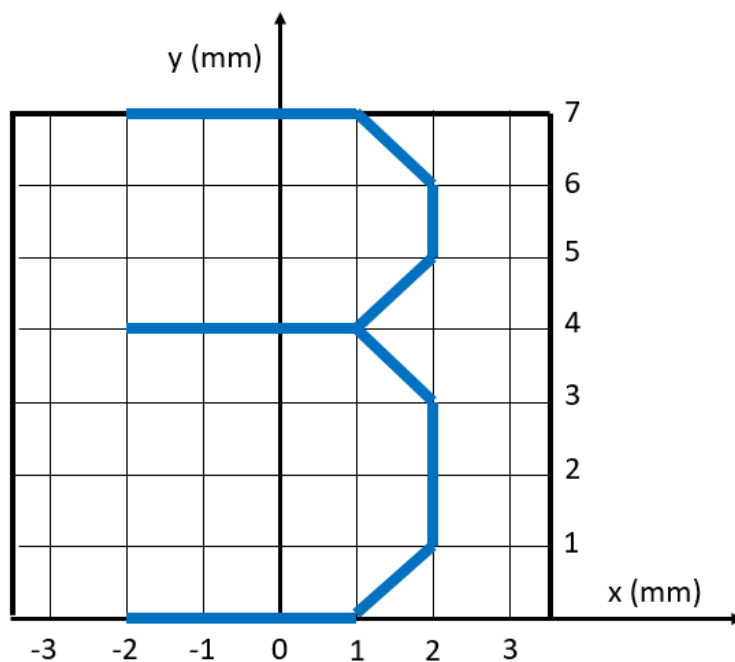


Ilustración 47: Diseño del número 3.

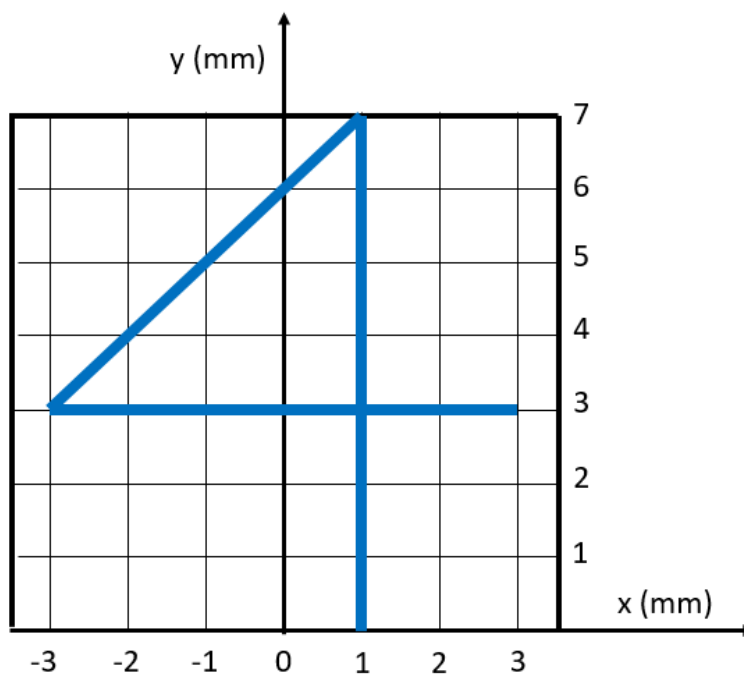


Ilustración 48: Diseño del número 4.

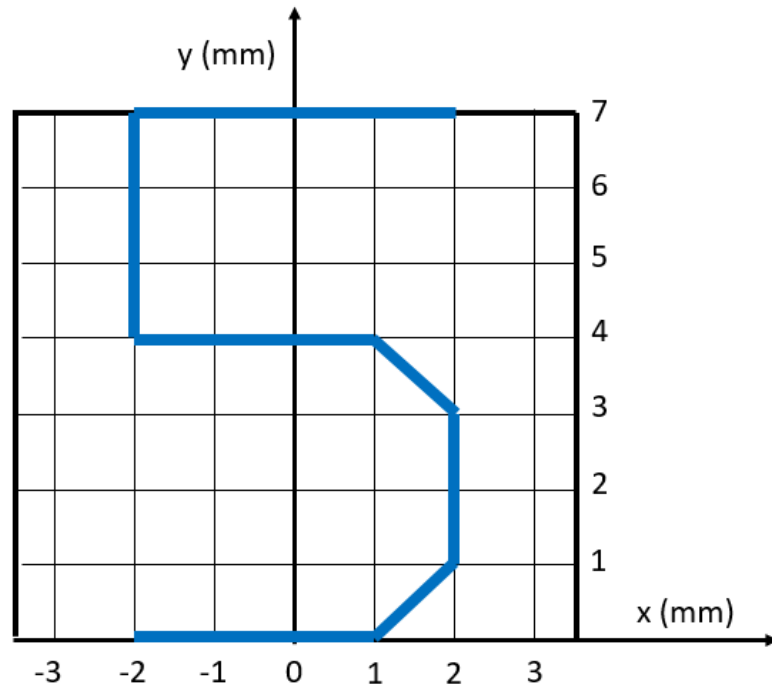


Ilustración 49: Diseño del número 5.

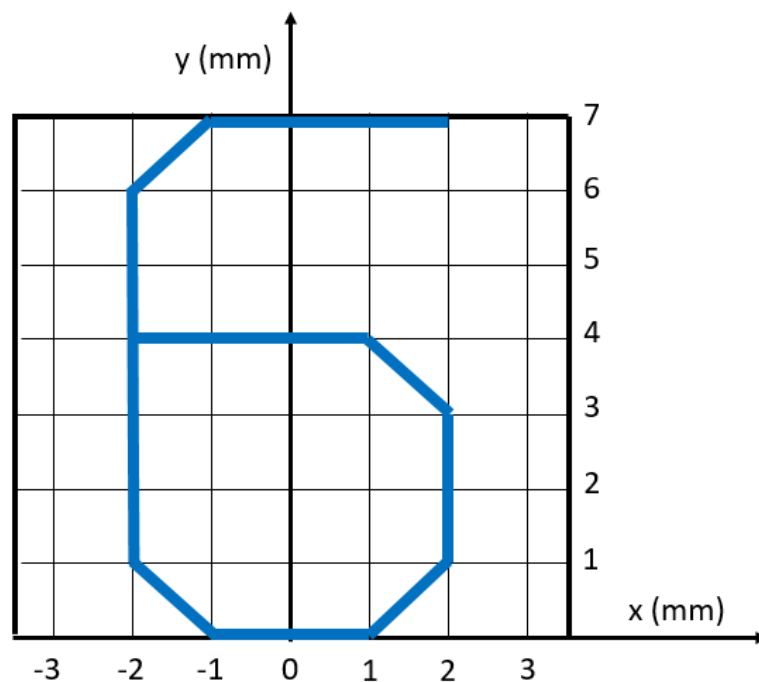


Ilustración 50: Diseño del número 6.

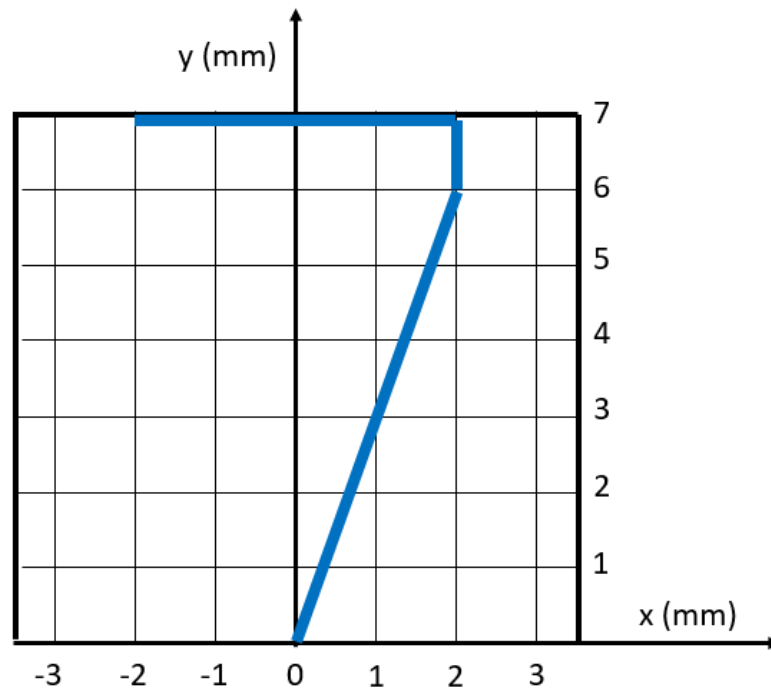


Ilustración 51: Diseño del número 7.

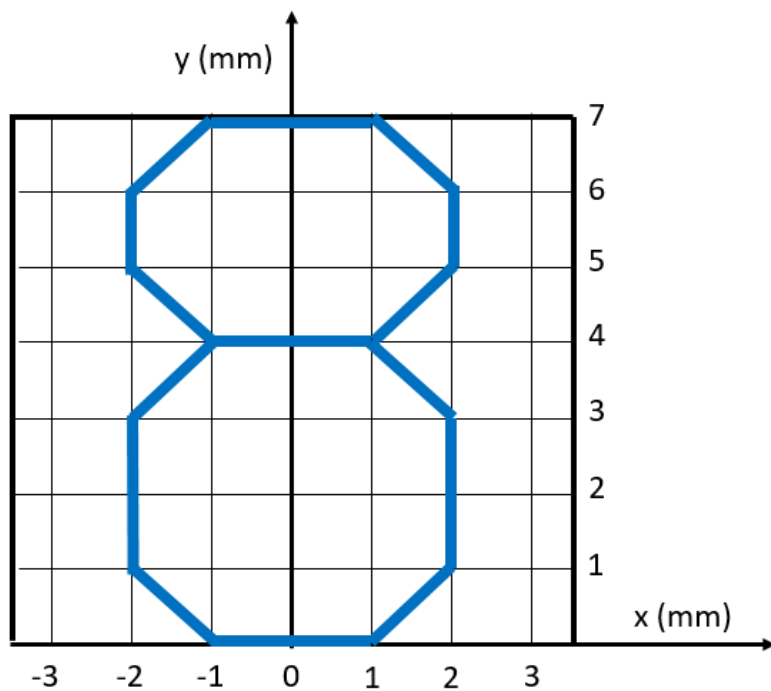


Ilustración 52: Diseño del número 8.

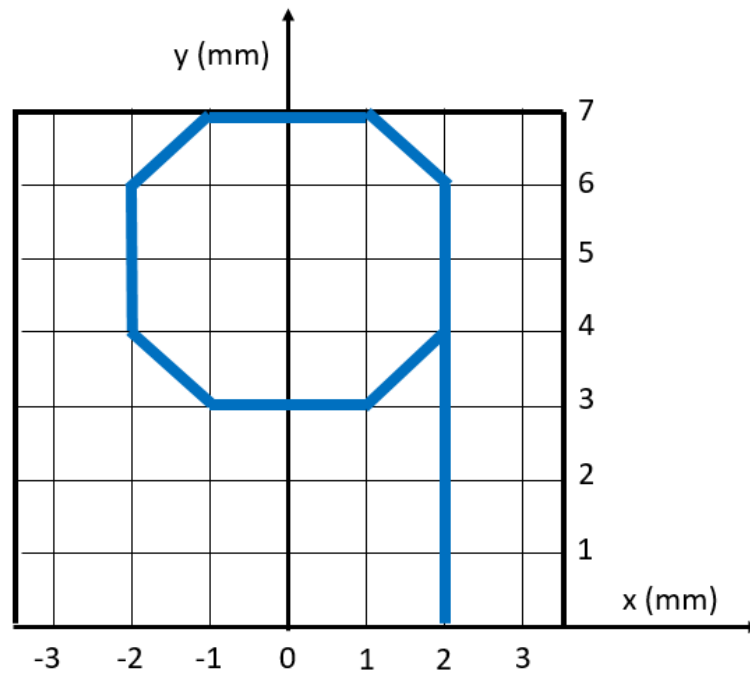
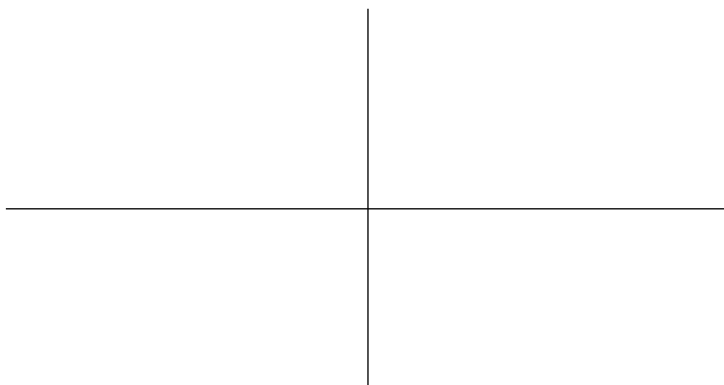
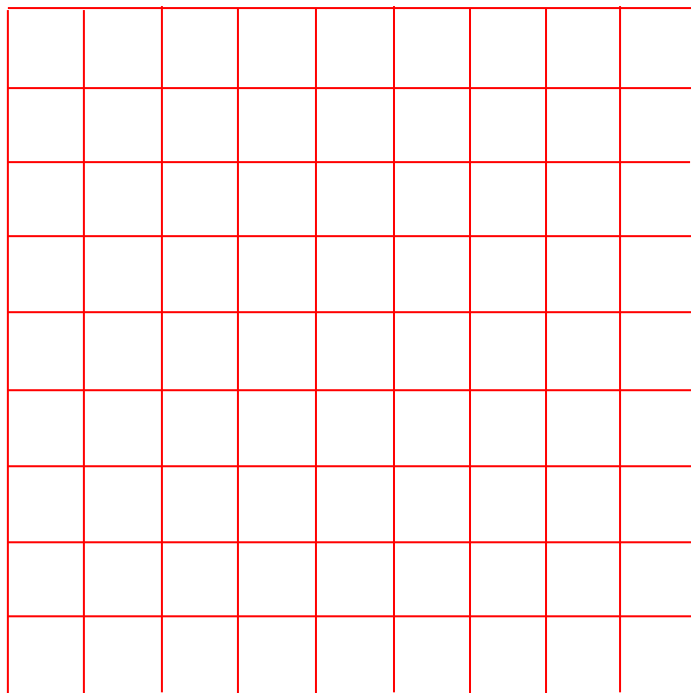


Ilustración 53: Diseño del número 9.

4. PLANTILLA PARA LA RESOLUCIÓN

Se encuentra en la página inmediatamente siguiente. Ya que la plantilla no puede contener título, encabezado o pie de página.



5. INSTRUCCIONES DE MONTAJE

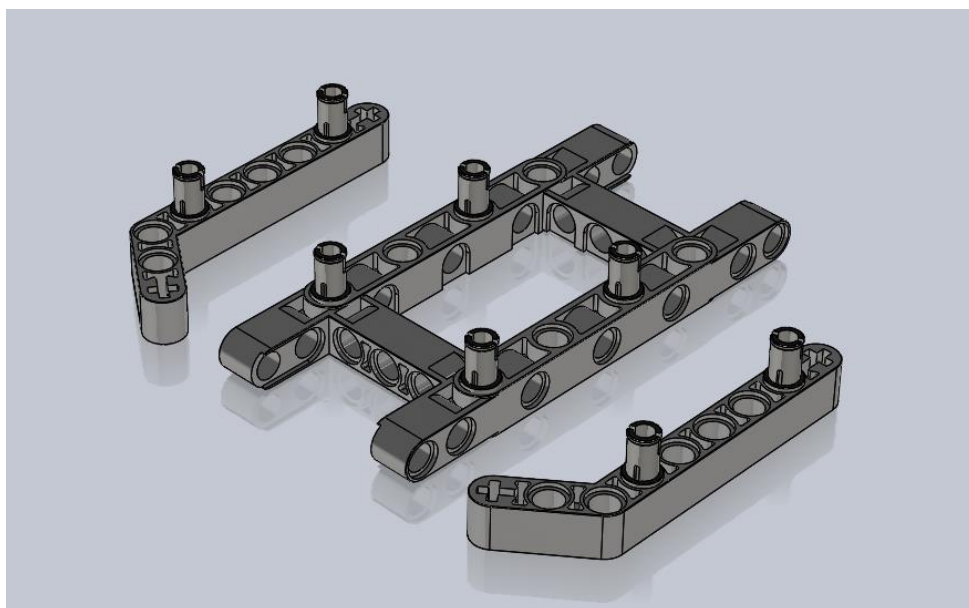


Ilustración 54: Montaje. Paso 1.

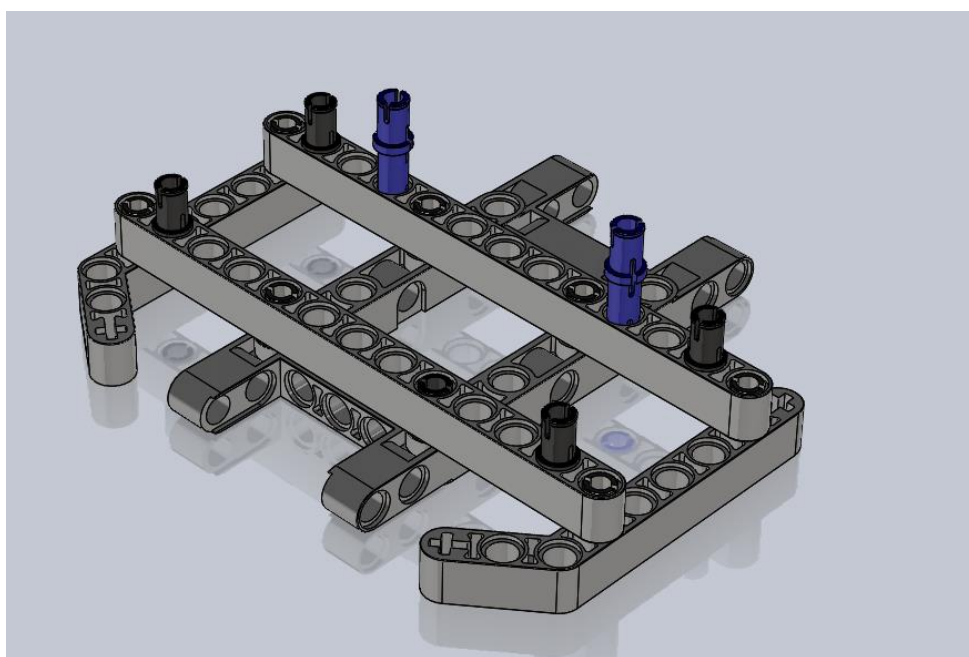


Ilustración 55: Montaje. Paso 2.

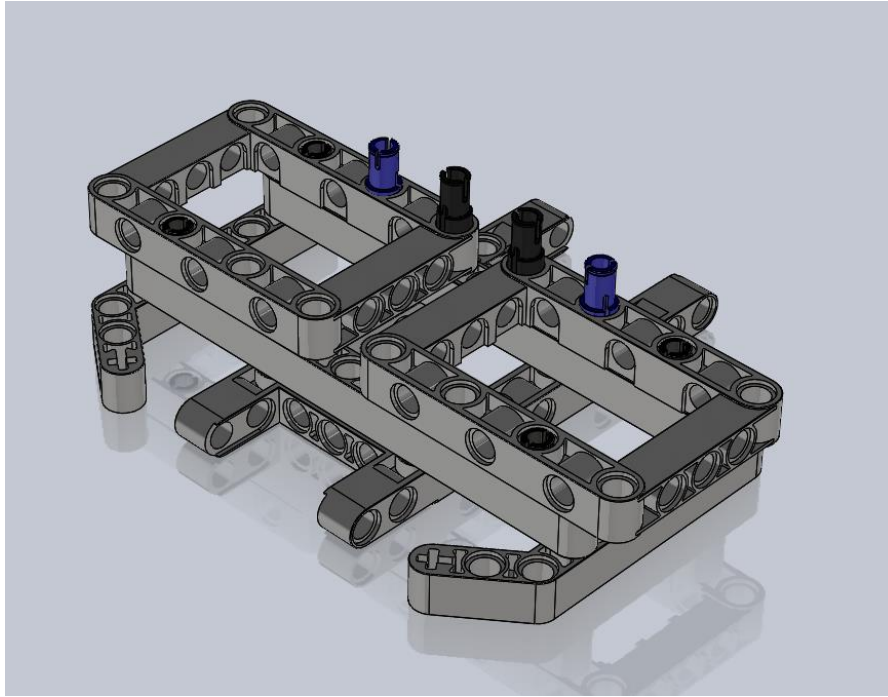


Ilustración 56: Montaje. Paso 3.

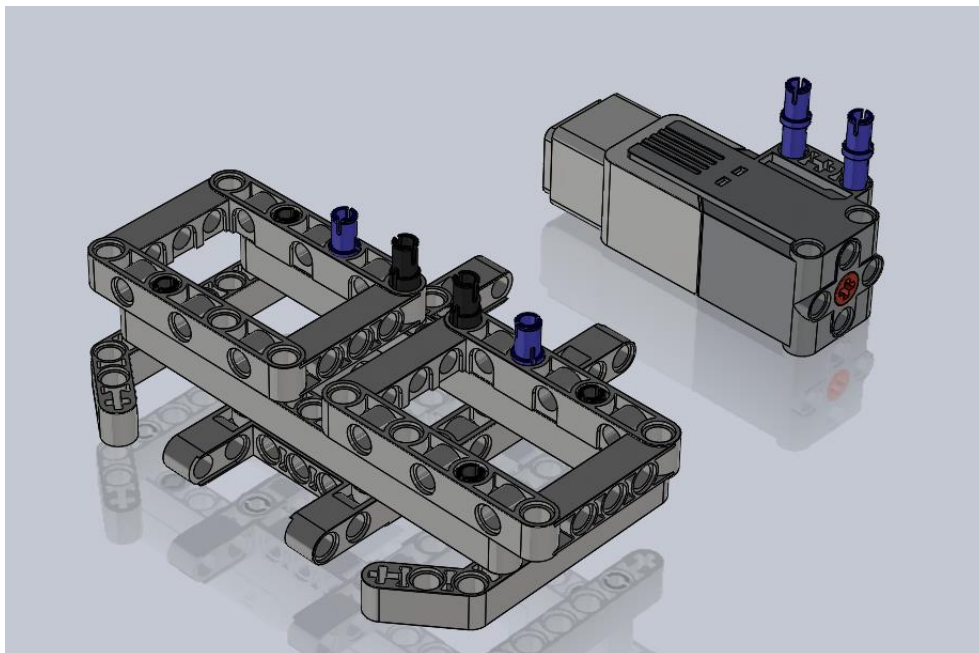


Ilustración 57: Montaje. Paso 4.

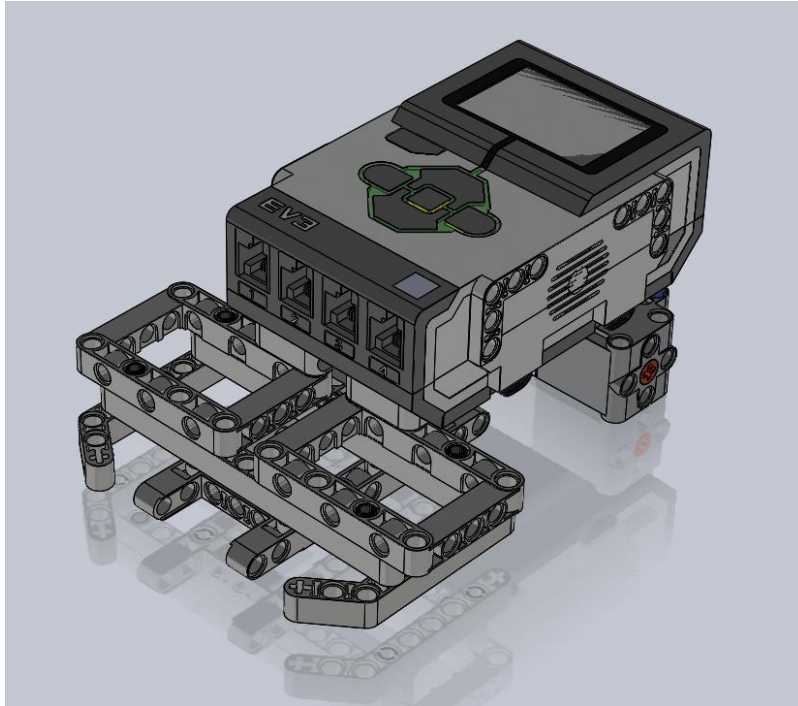


Ilustración 58: Montaje. Paso 5.

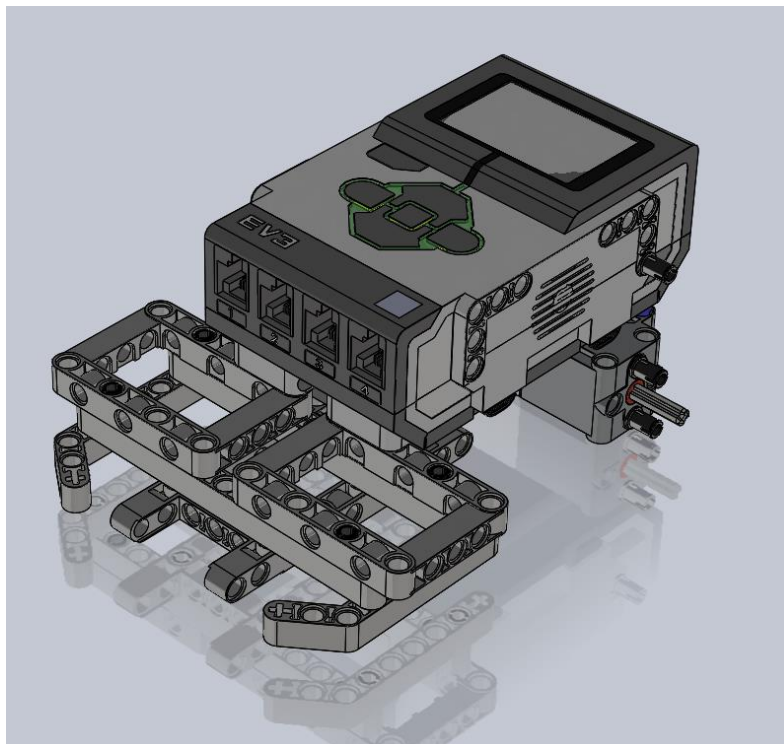


Ilustración 59: Montaje. Paso 6.

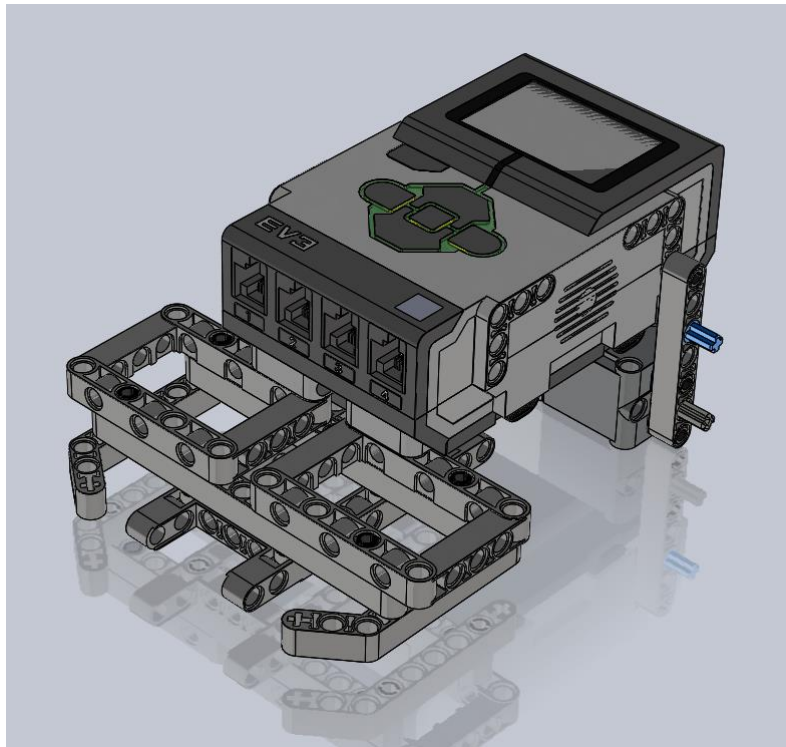


Ilustración 60: Montaje. Paso 7.

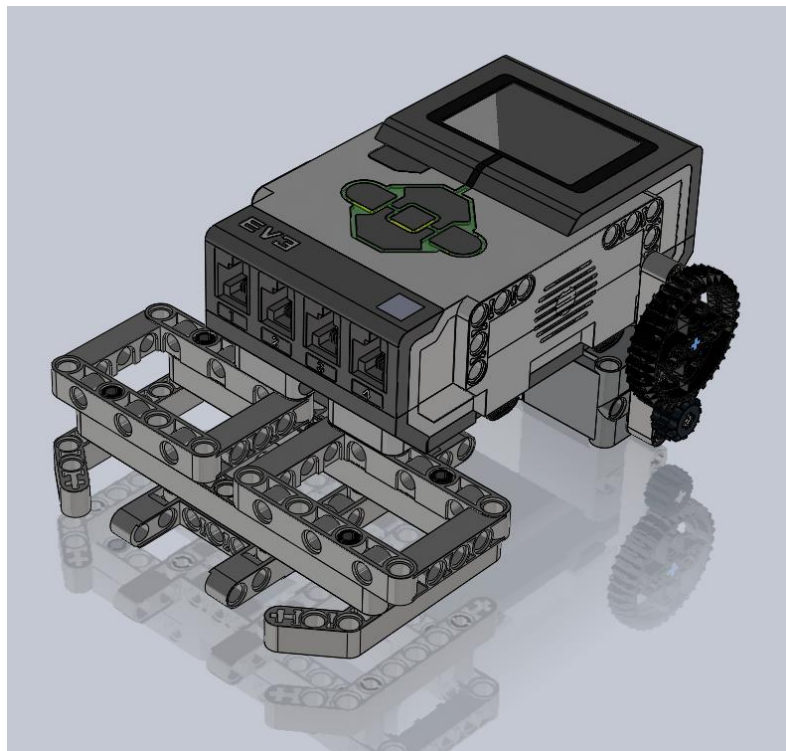


Ilustración 61: Montaje. Paso 8.

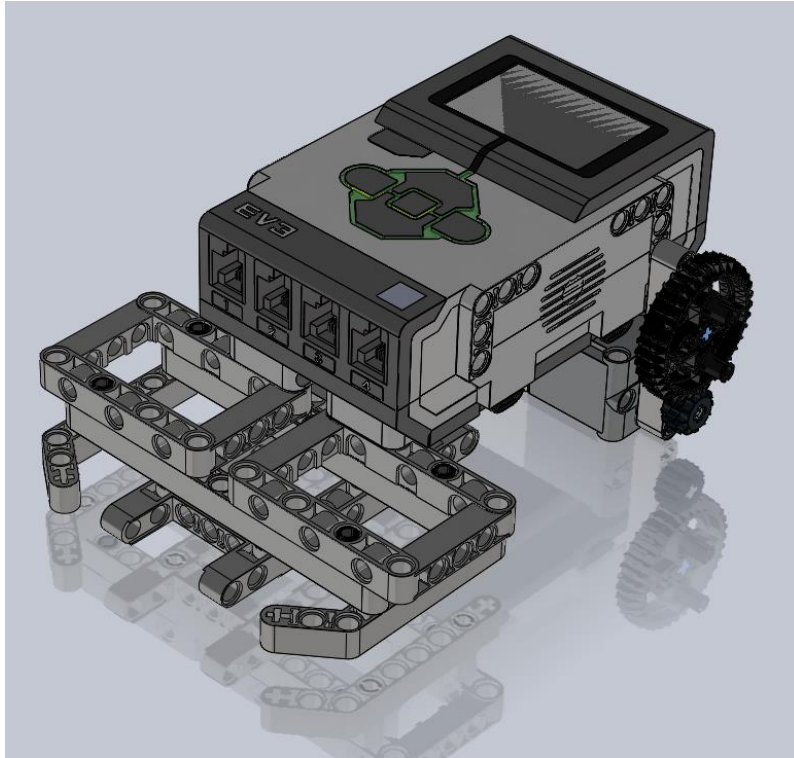


Ilustración 62: Montaje. Paso 9.

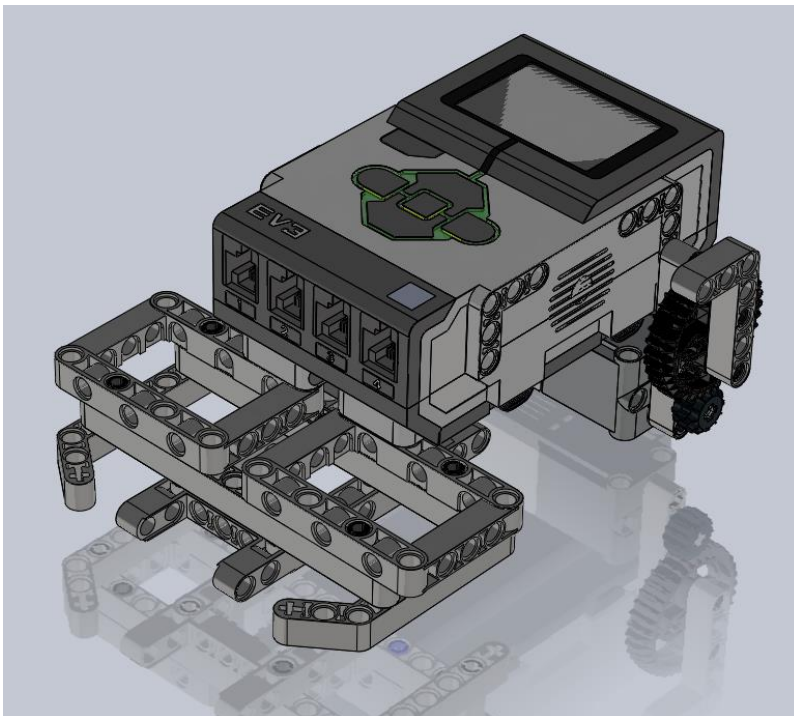


Ilustración 63: Montaje. Paso 10.

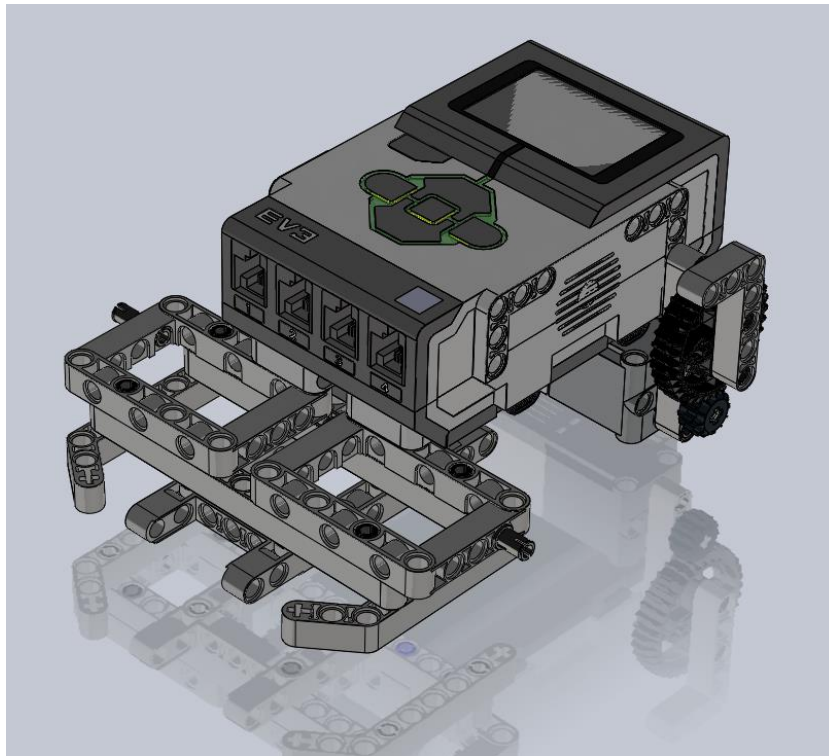


Ilustración 64: Montaje. Paso 11.

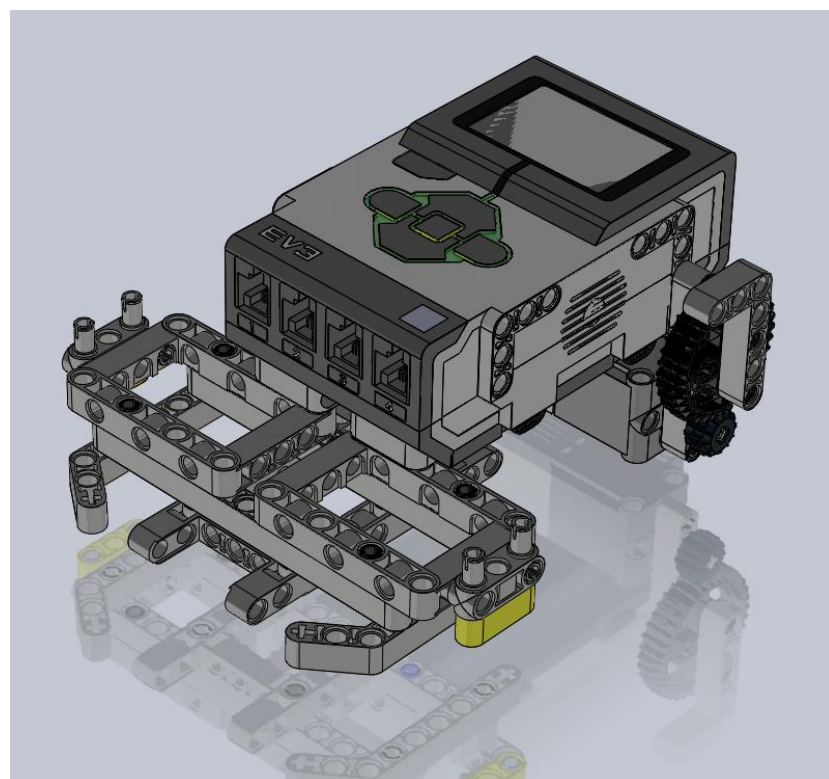


Ilustración 65: Montaje. Paso 12.

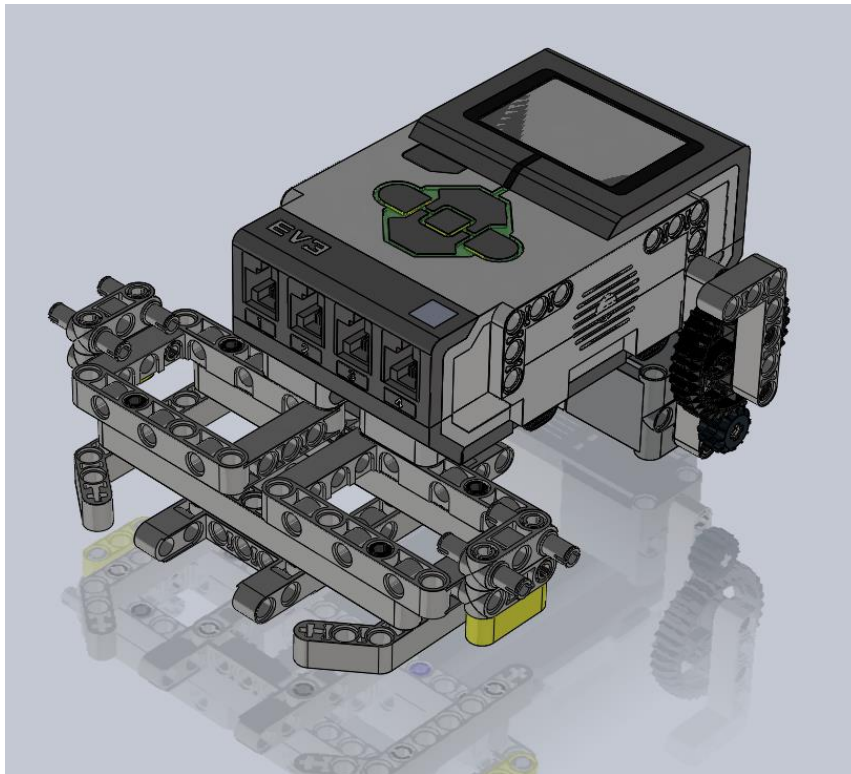


Ilustración 66: Montaje. Paso 13.

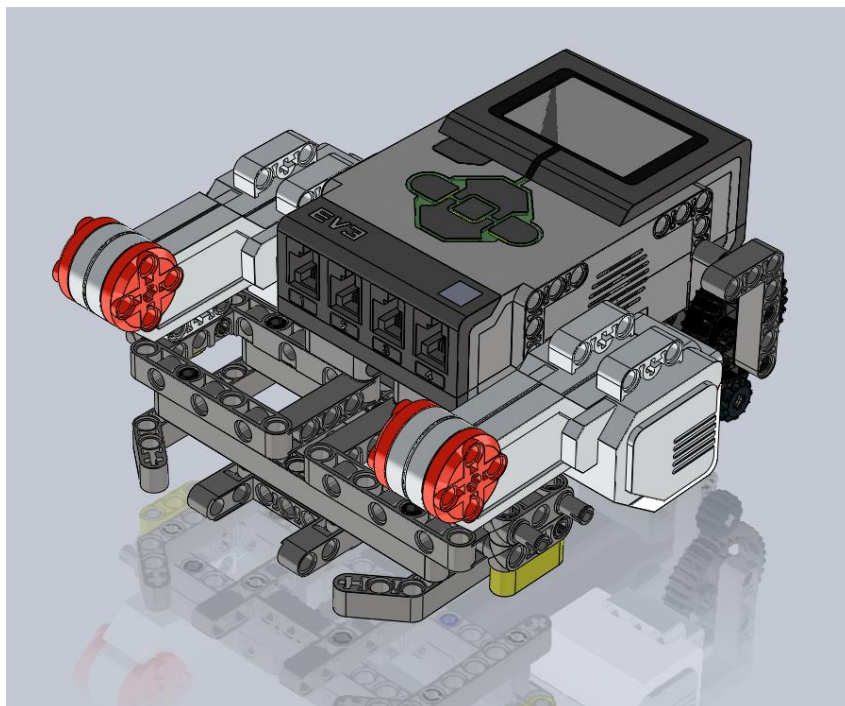


Ilustración 67: Montaje. Paso 14.

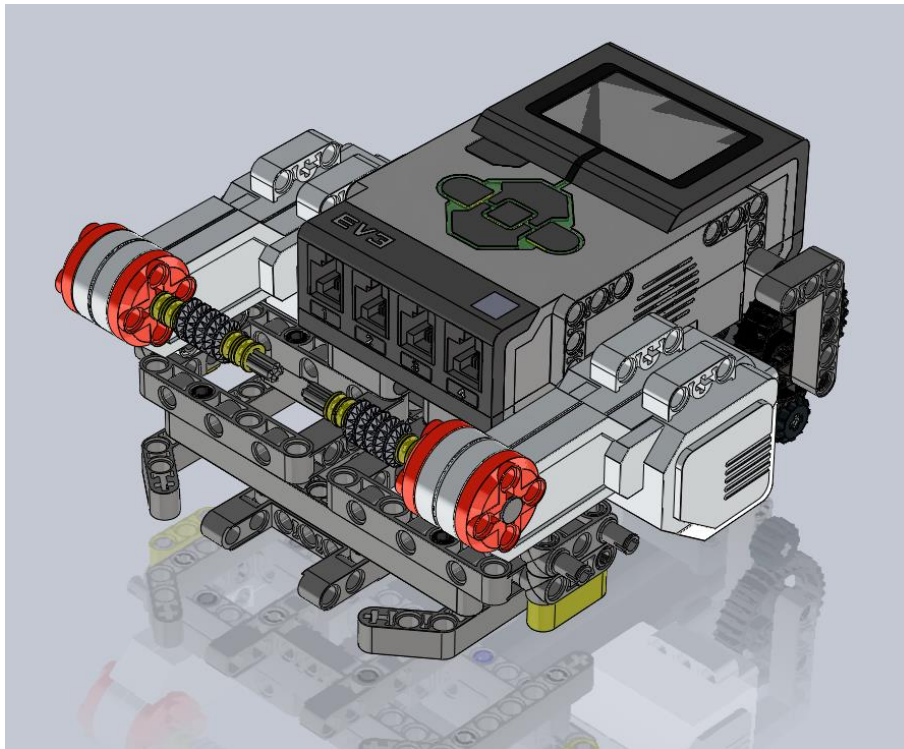


Ilustración 68: Montaje. Paso 15.

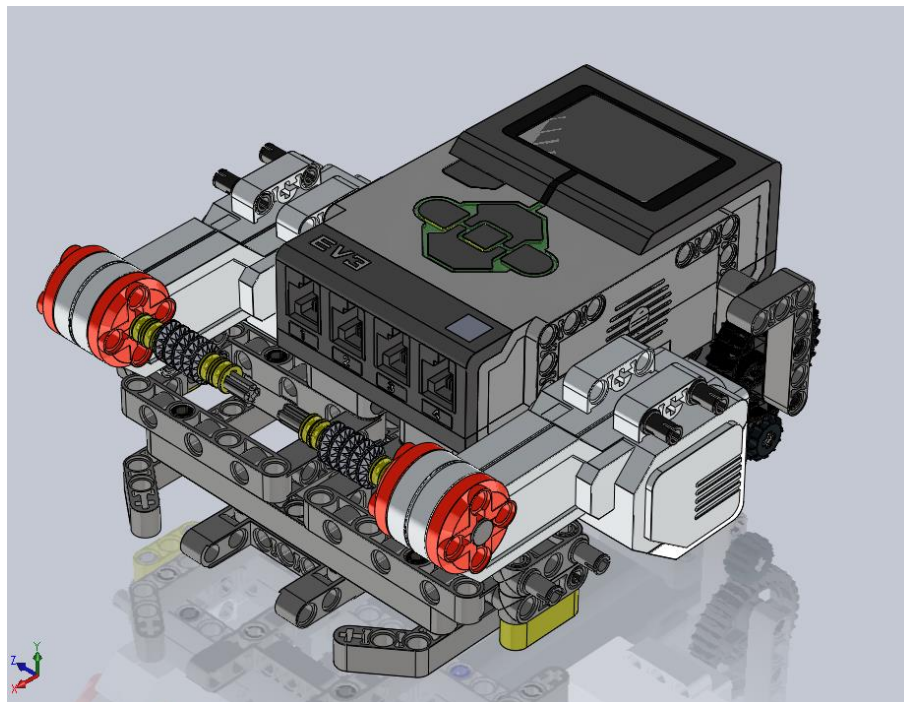


Ilustración 69: Montaje. Paso 16.

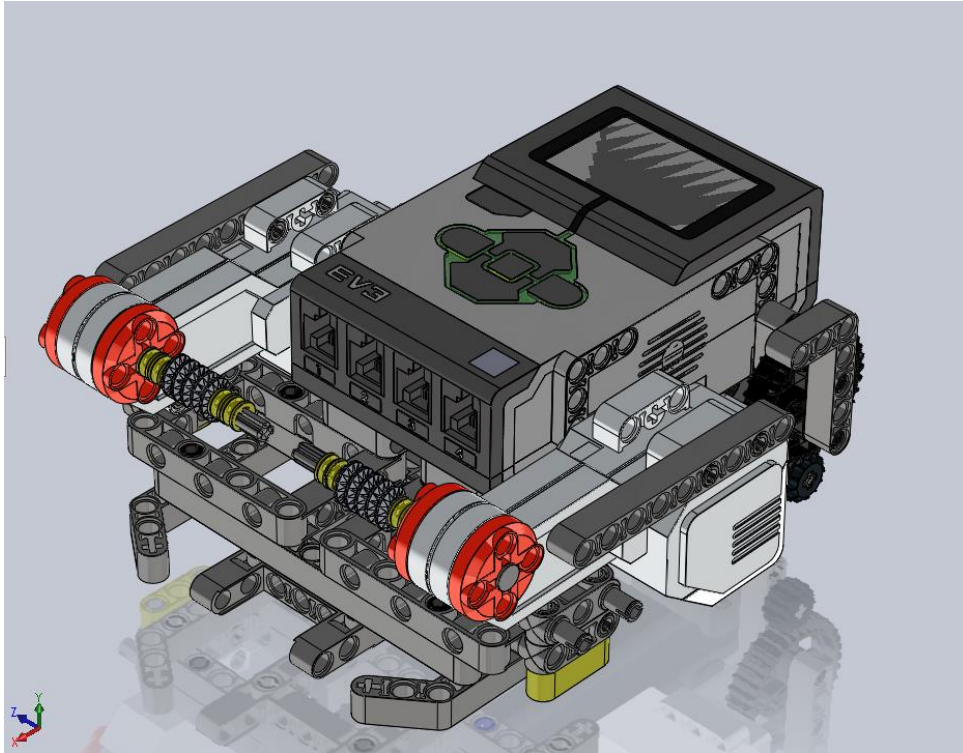


Ilustración 70: Montaje. Paso 17.

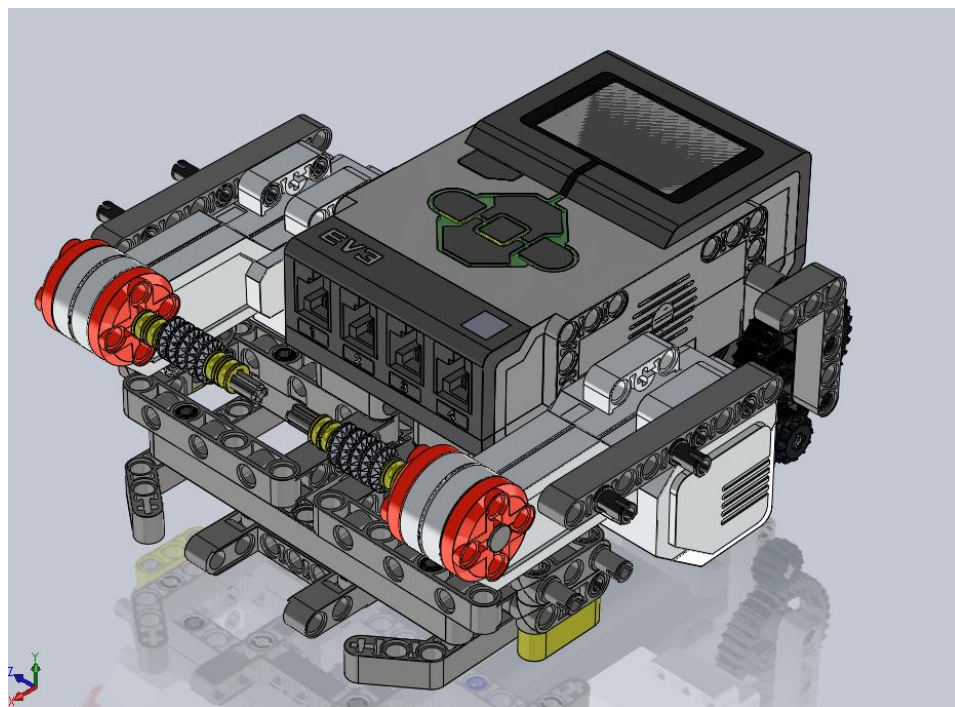


Ilustración 71: Montaje. Paso 18.

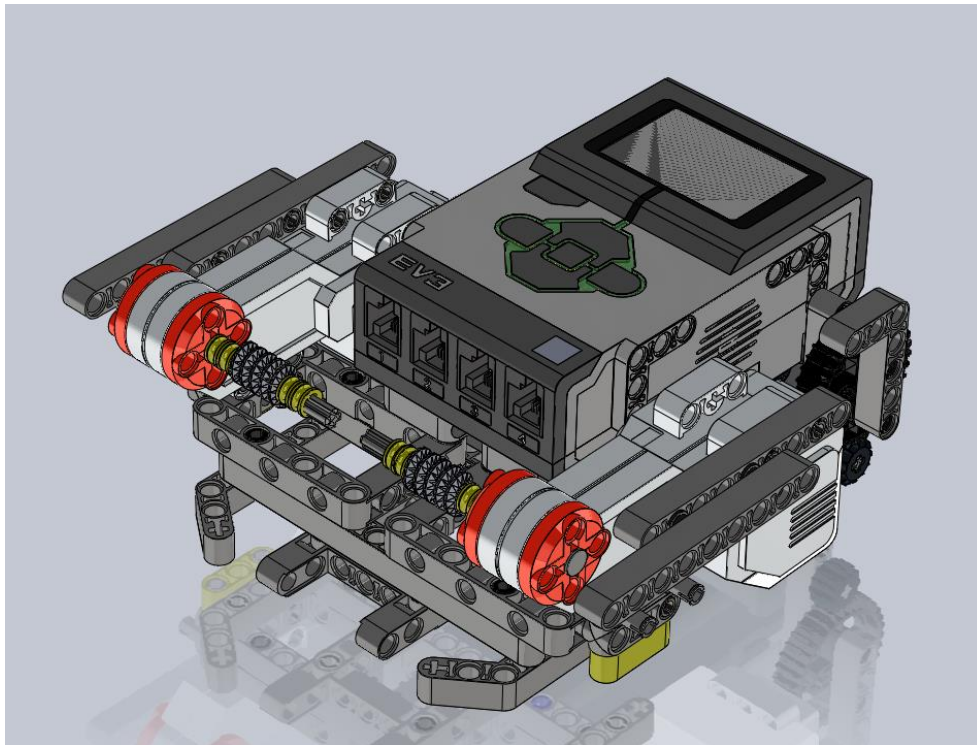


Ilustración 72: Montaje. Paso 19.

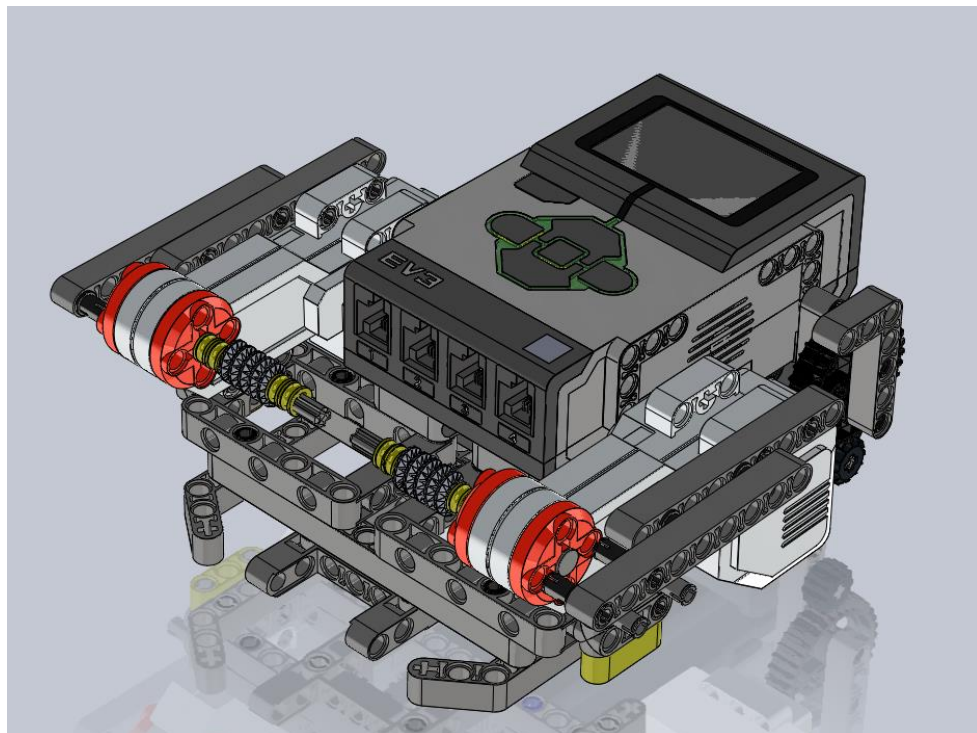


Ilustración 73: Montaje. Paso 20.

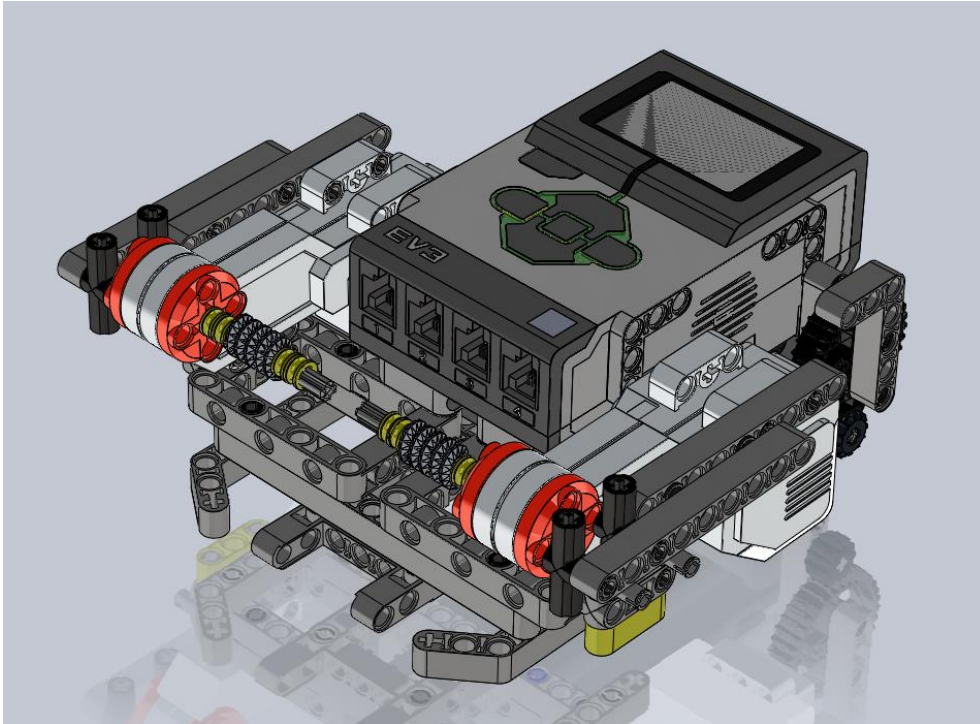


Ilustración 74: Montaje. Paso 21.

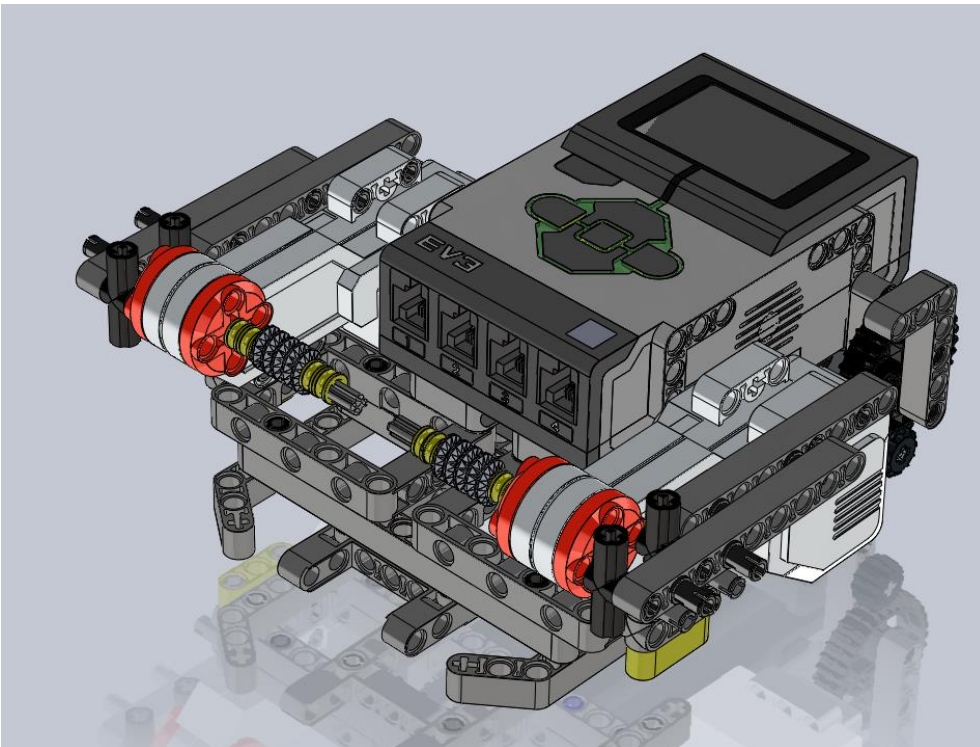


Ilustración 75: Montaje. Paso 22.

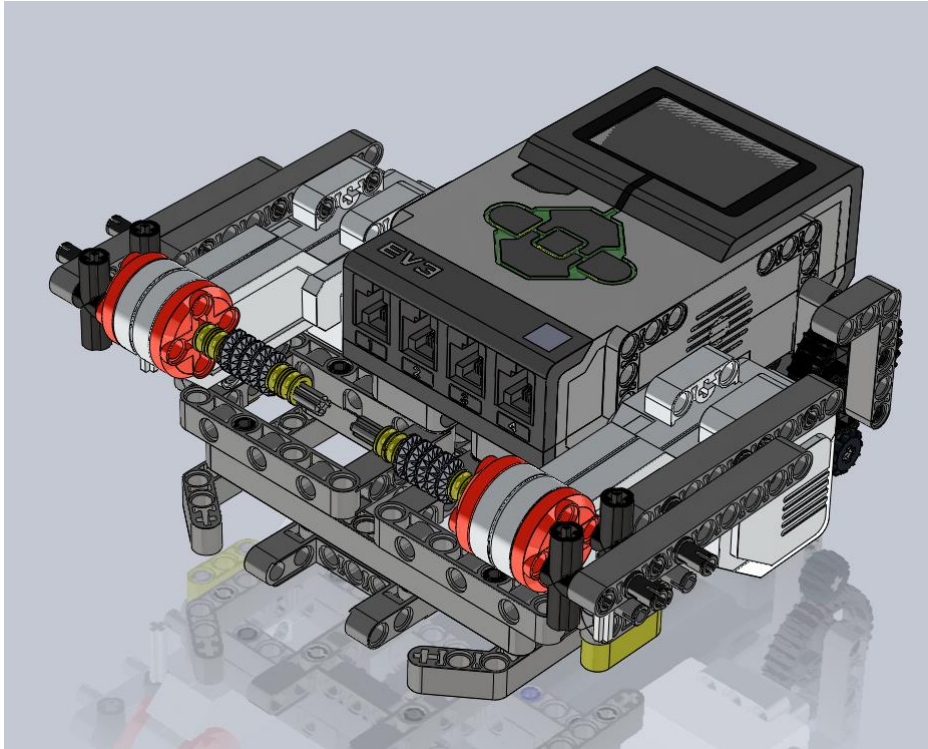


Ilustración 76: Montaje. Paso 23.

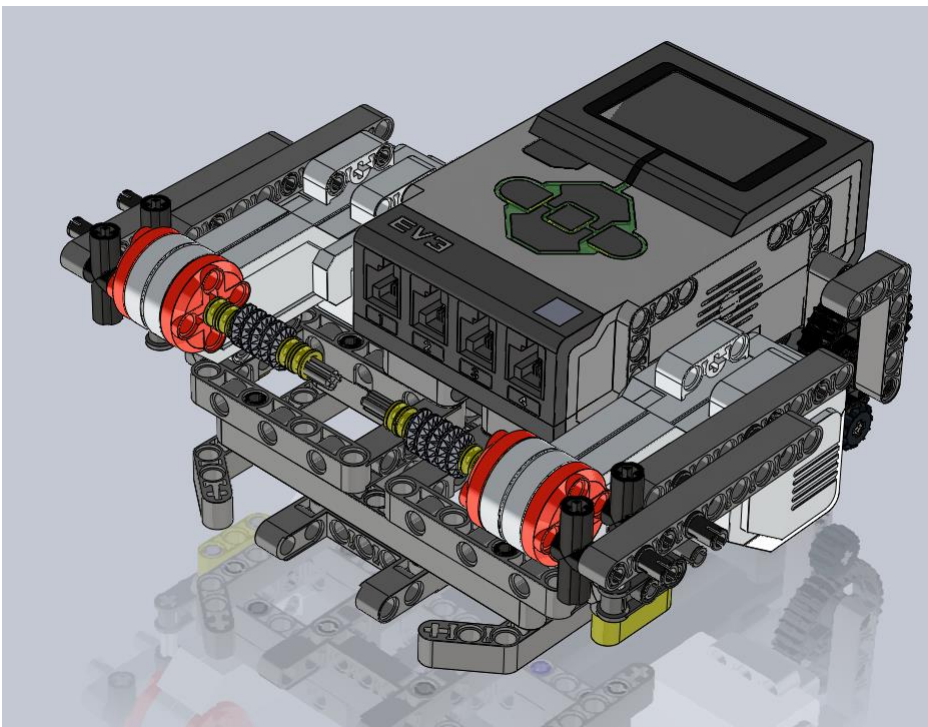


Ilustración 77: Montaje. Paso 24.

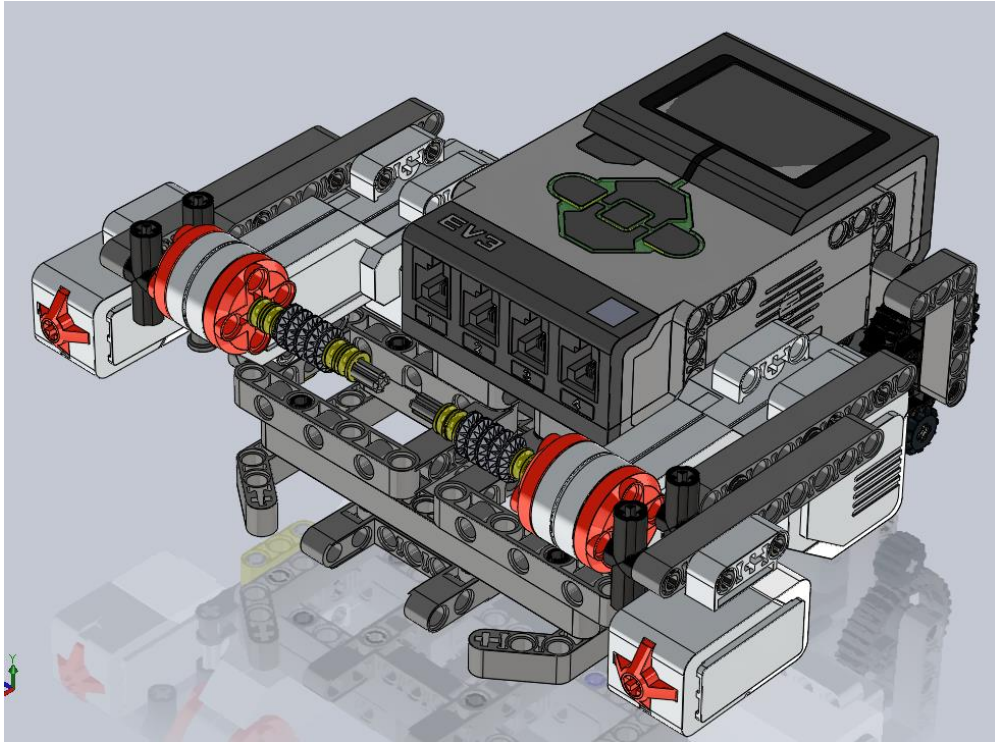


Ilustración 78: Montaje. Paso 25.

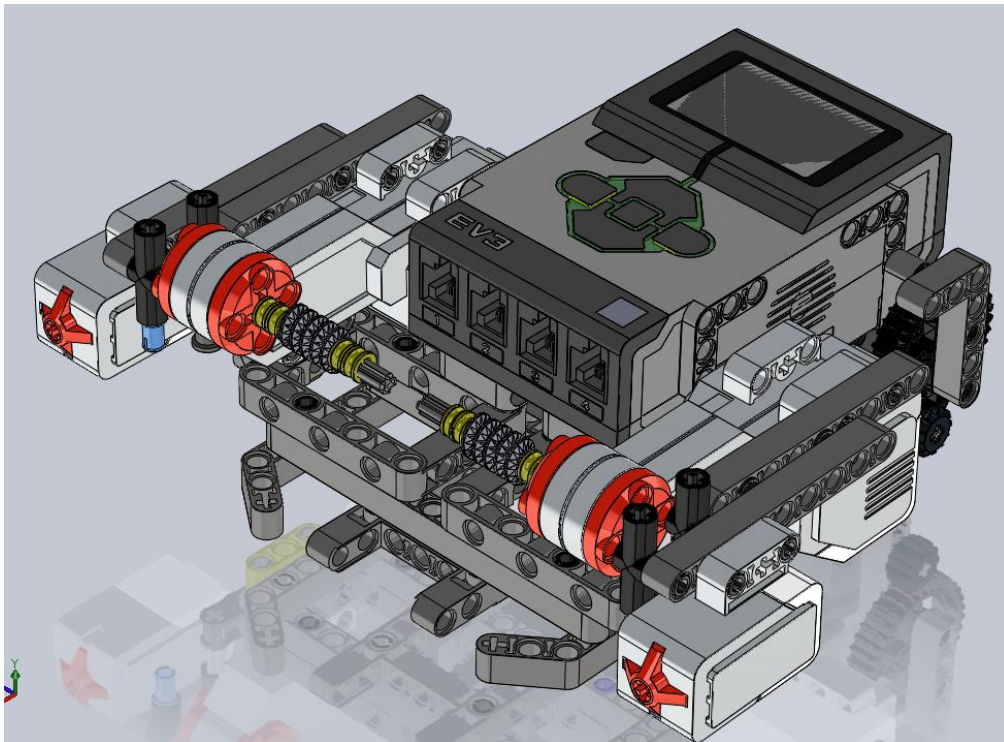


Ilustración 79: Montaje. Paso 26.

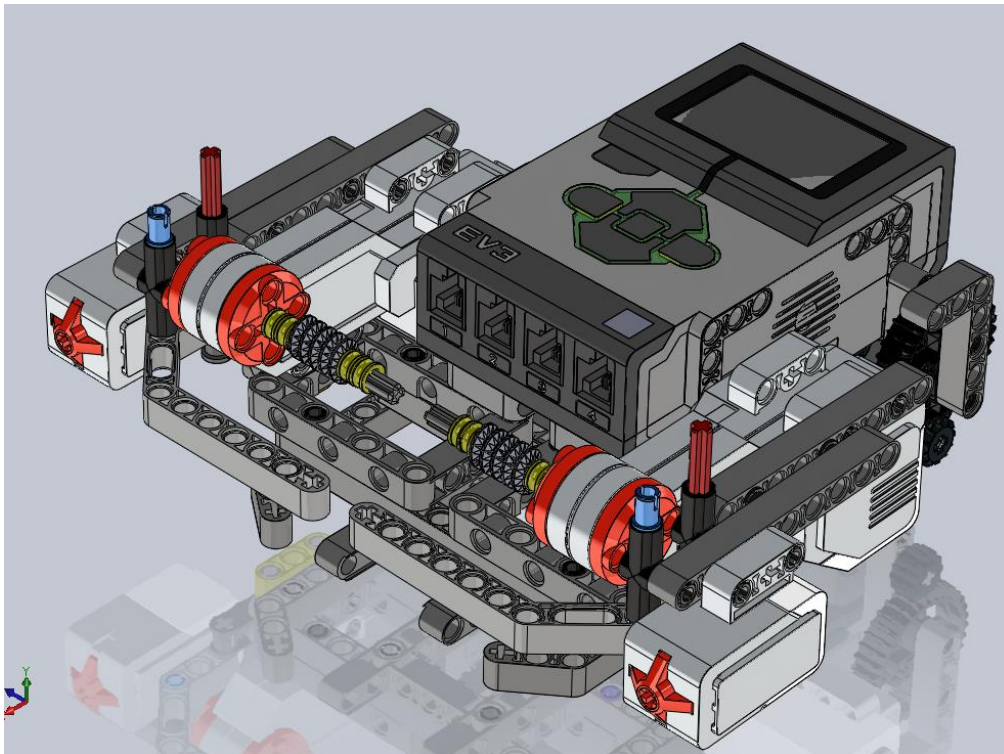


Ilustración 80: Montaje. Paso 27.

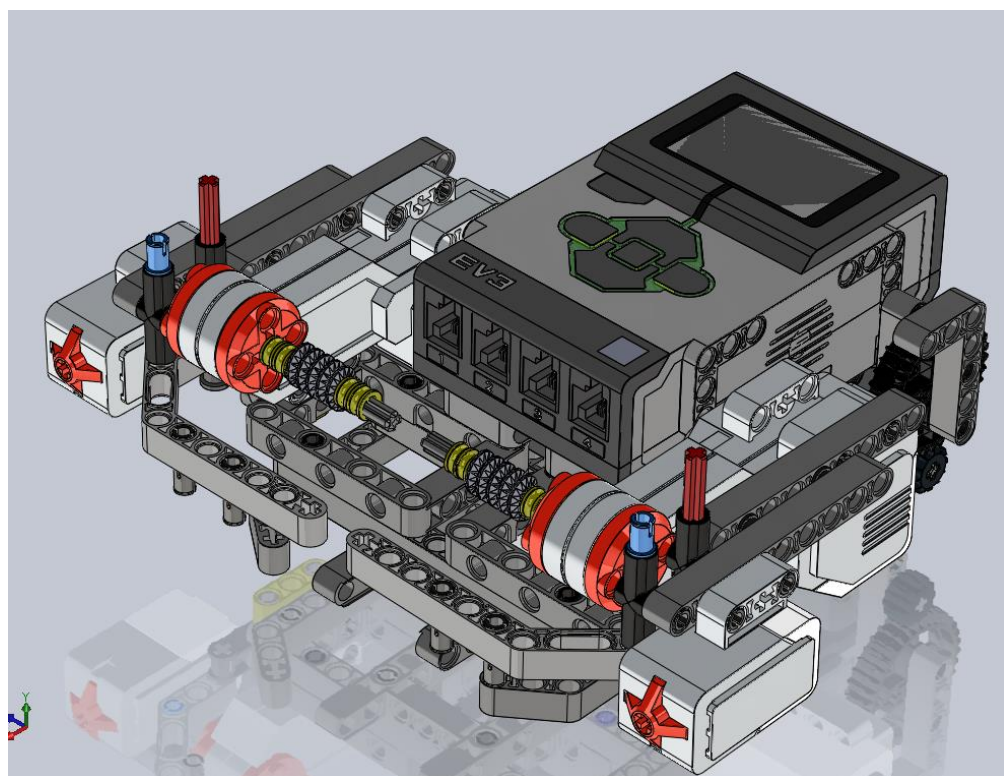


Ilustración 81: Montaje. Paso 28.

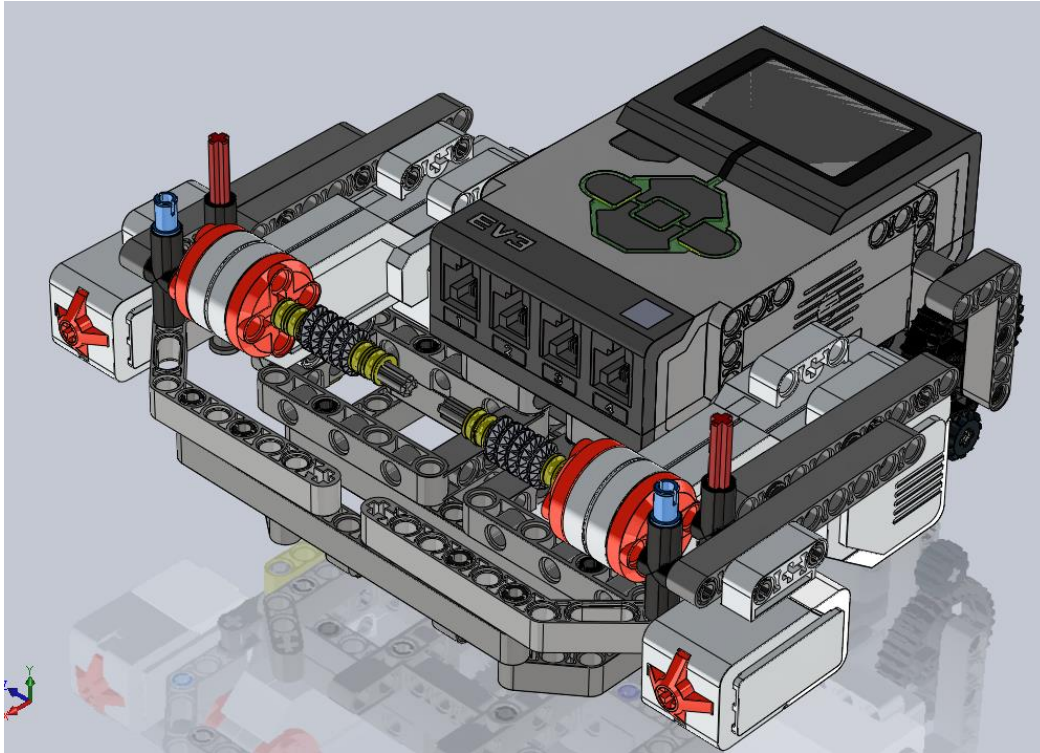


Ilustración 82: Montaje. Paso 29.

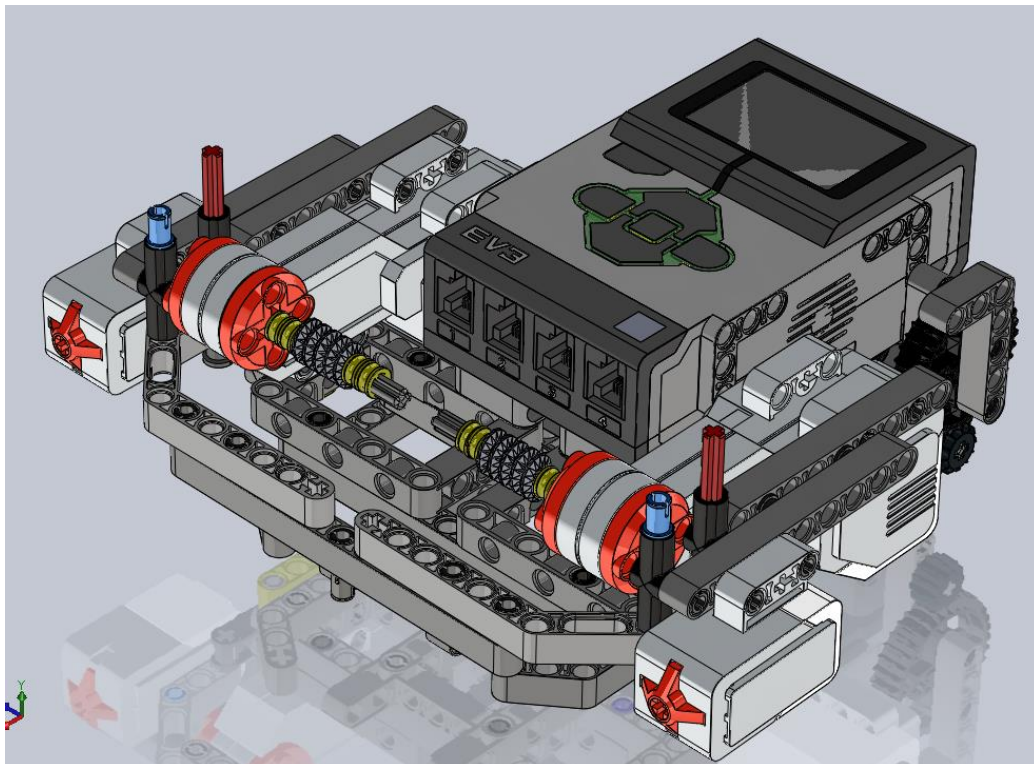


Ilustración 83: Montaje. Paso 30.

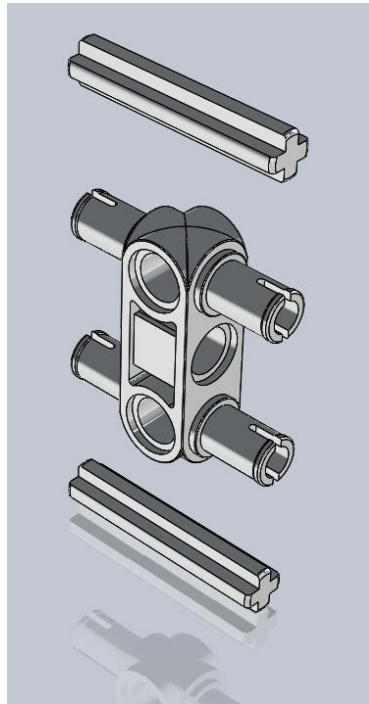


Ilustración 84: Montaje. Paso 31.

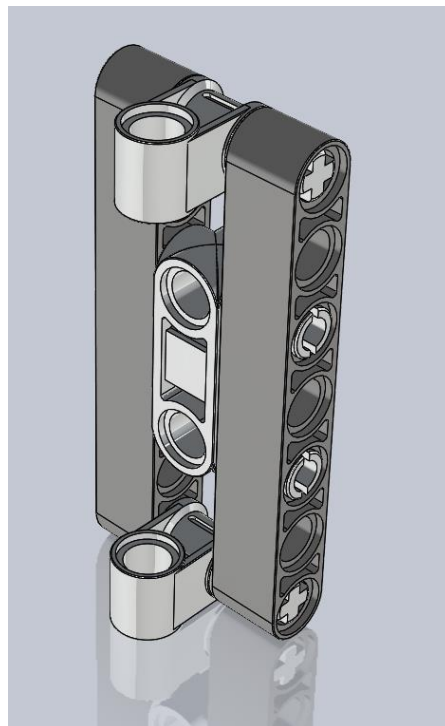


Ilustración 85: Montaje. Paso 32.

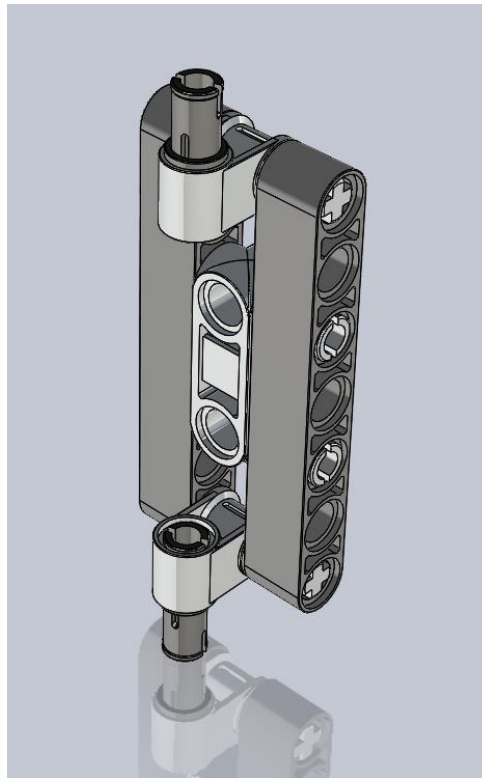


Ilustración 86: Montaje. Paso33.

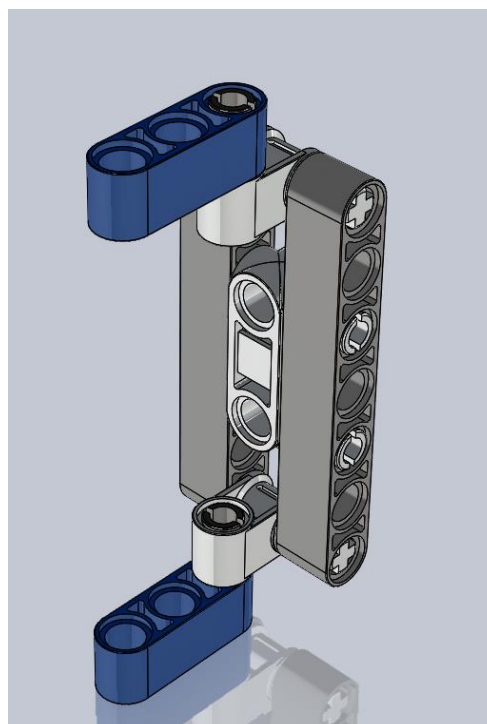


Ilustración 87: Montaje. Paso 34.

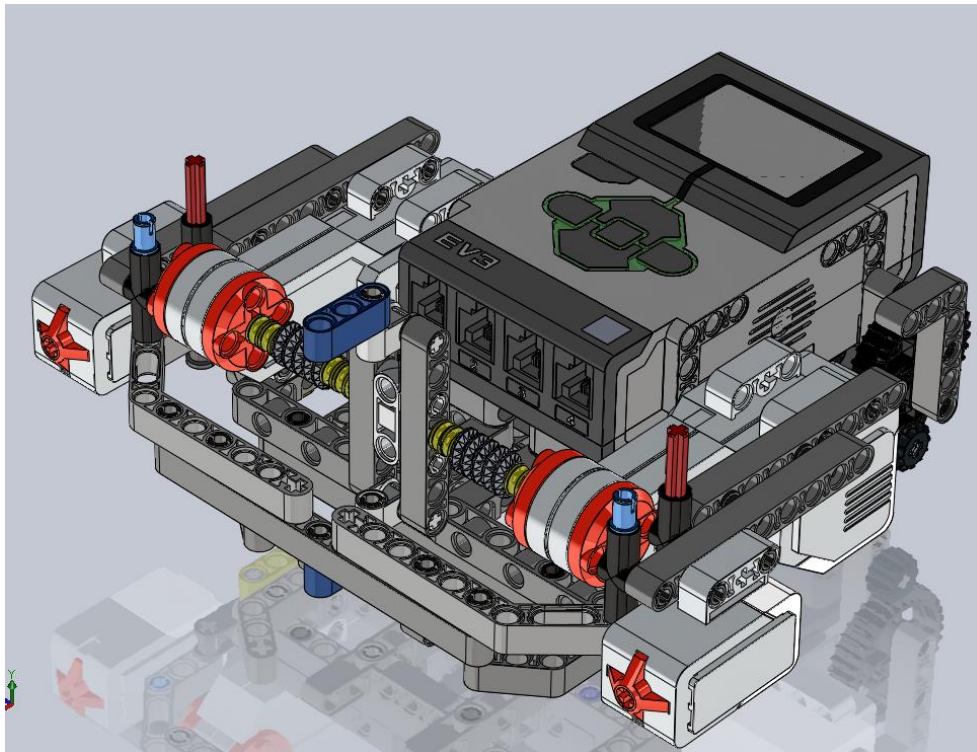


Ilustración 88: Montaje. Paso 35.

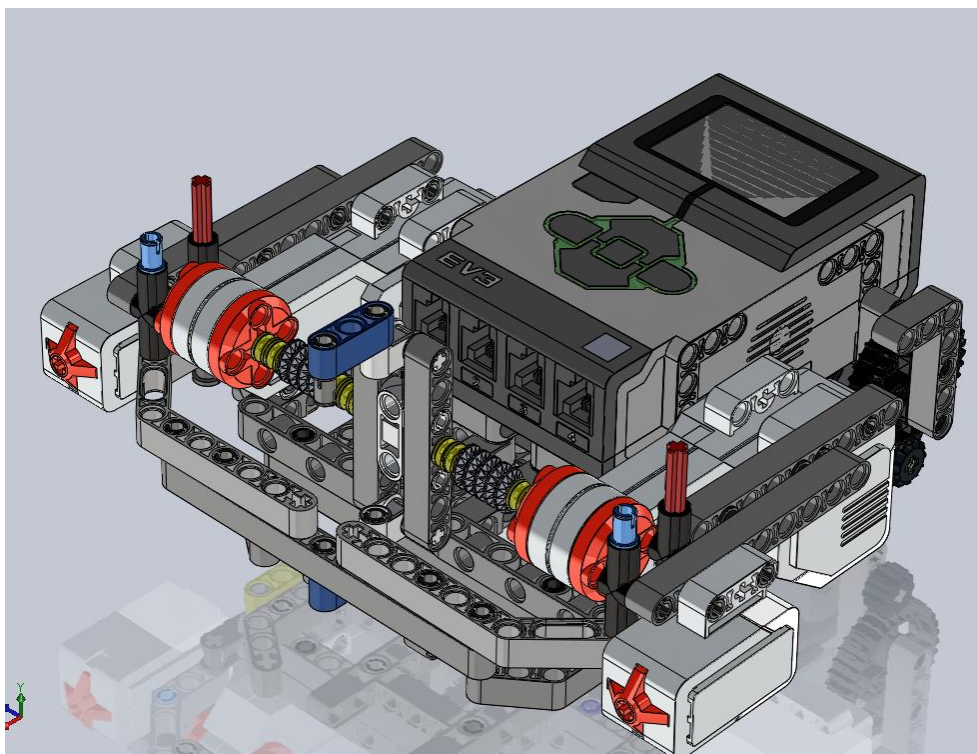


Ilustración 89: Montaje. Paso 36.

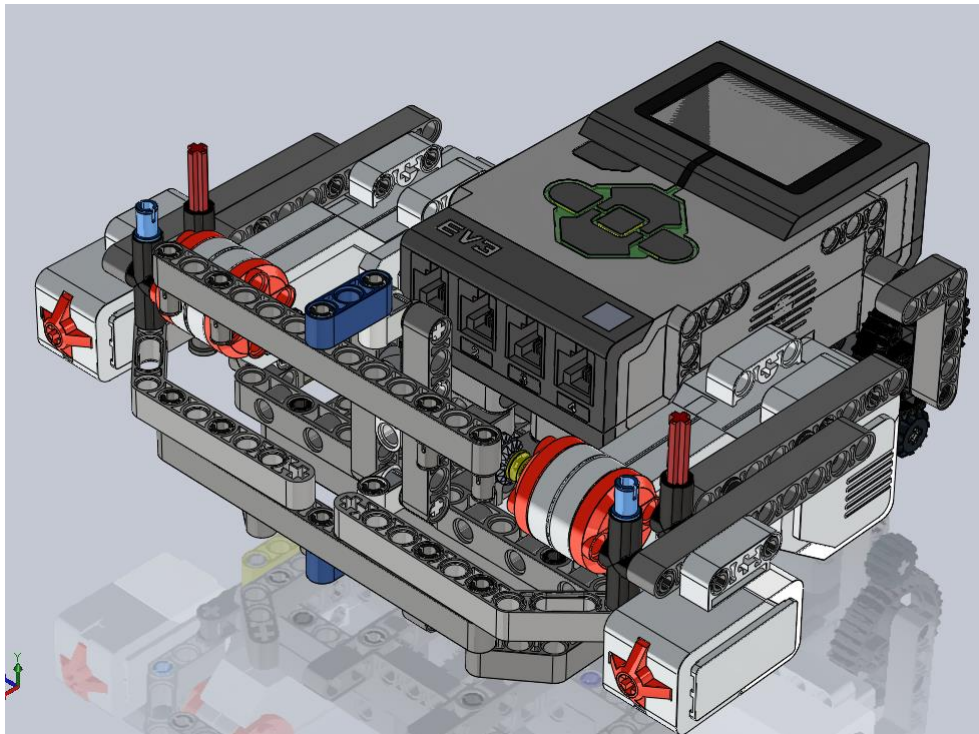


Ilustración 90: Montaje. Paso 37.

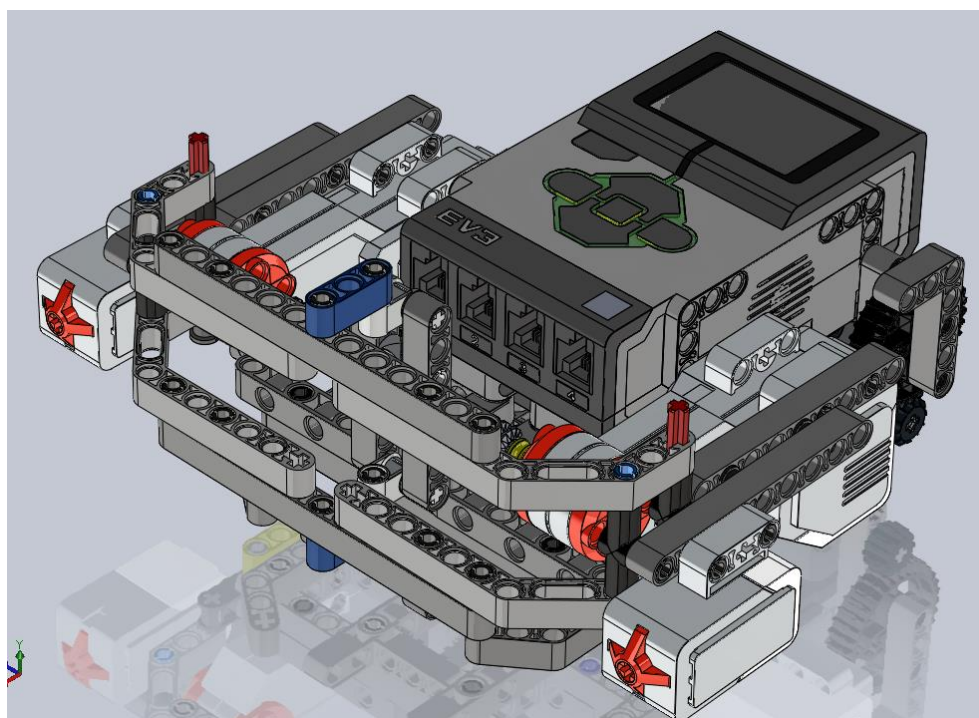


Ilustración 91: Montaje. Paso 38.

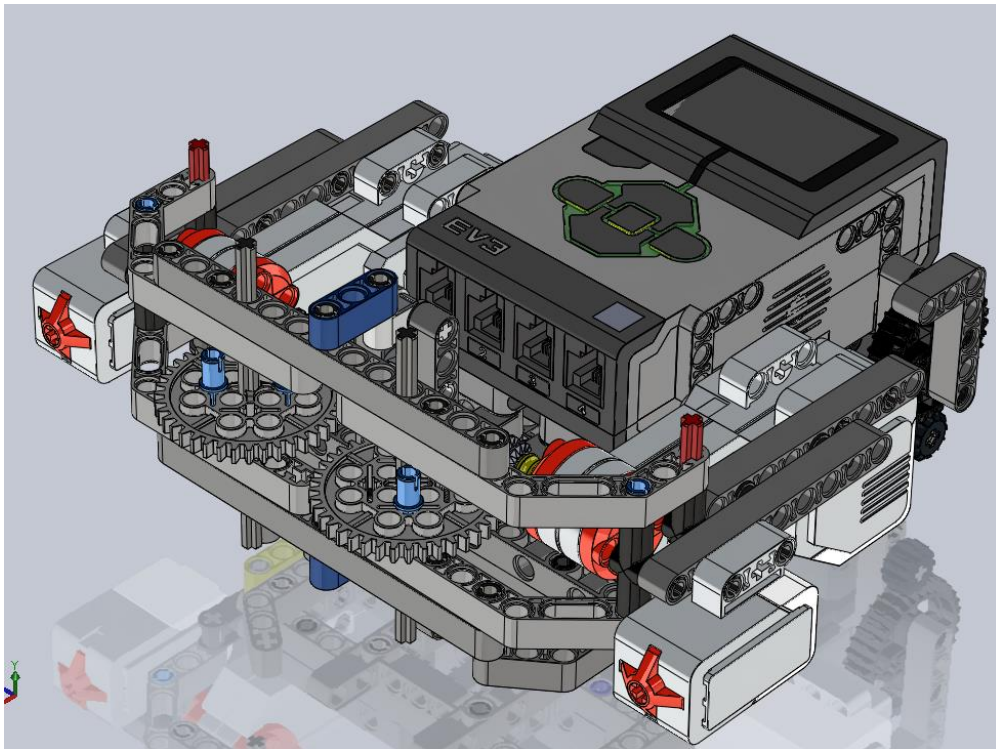


Ilustración 92: Montaje. Paso 39.

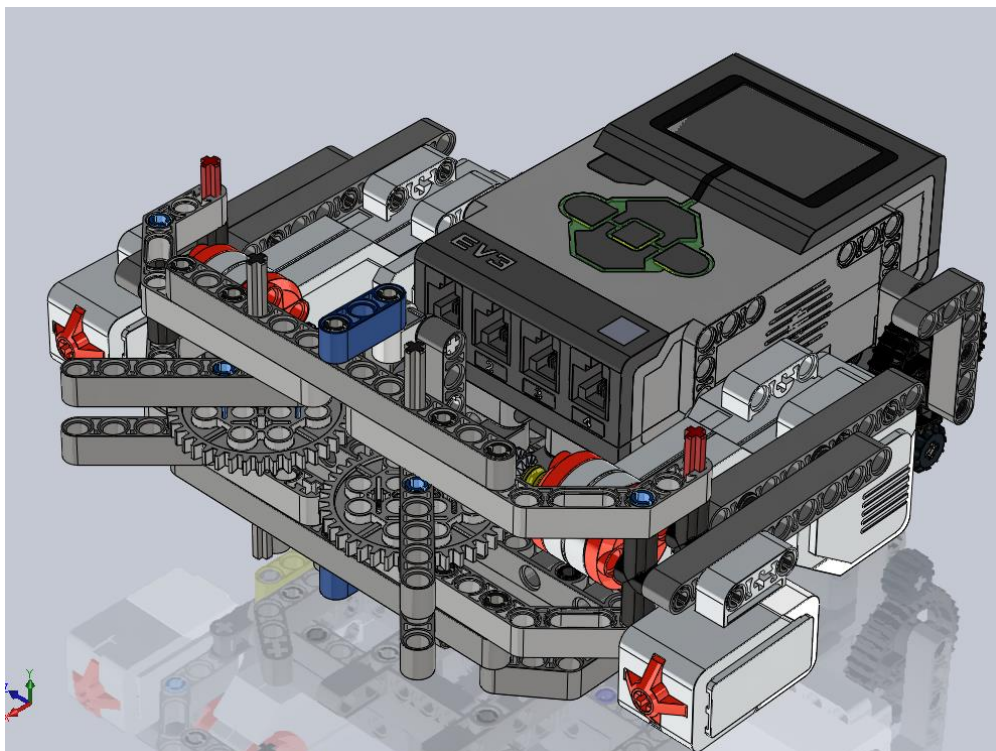


Ilustración 93: Montaje. Paso 40.

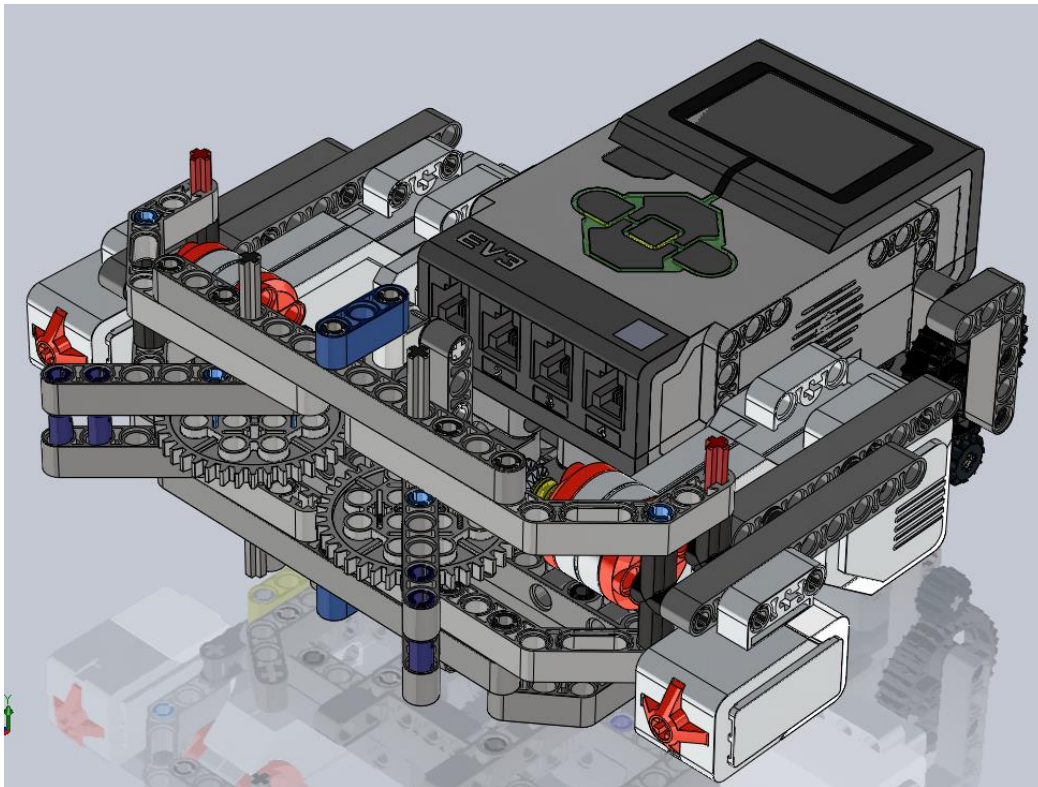


Ilustración 94: Montaje. Paso 41.

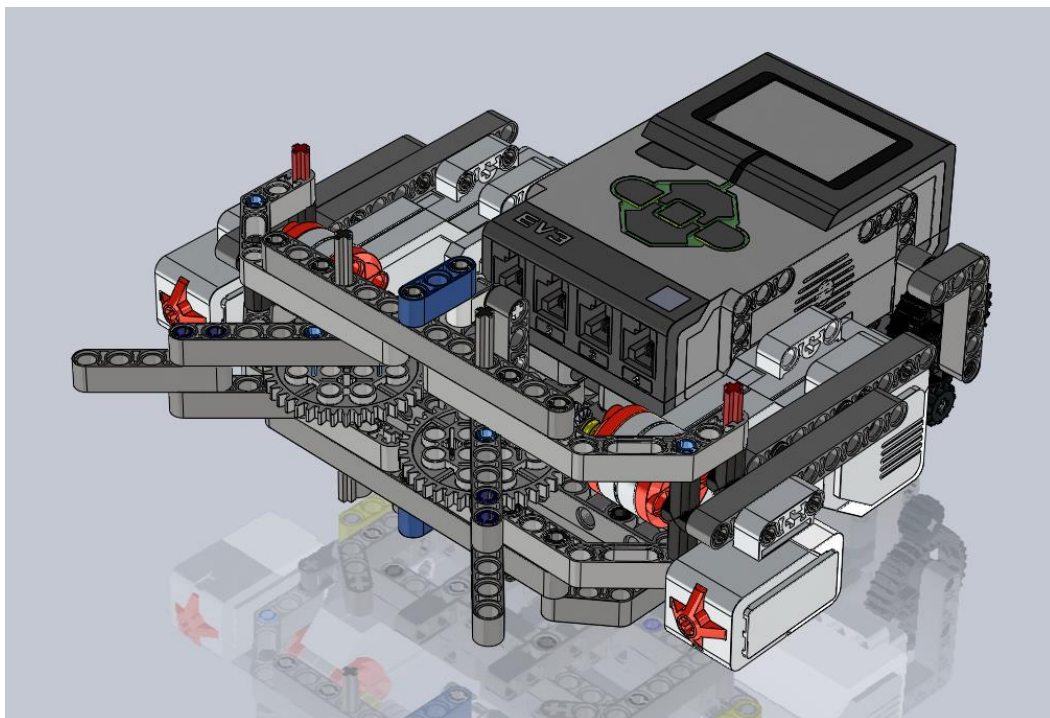


Ilustración 95: Montaje. Paso 42.

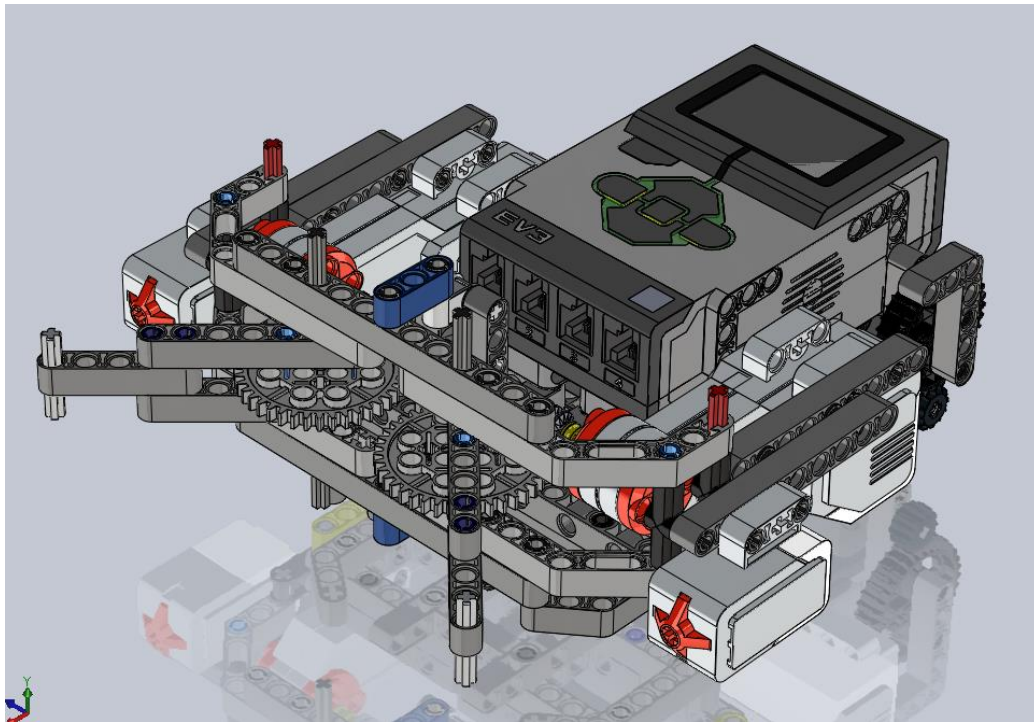


Ilustración 96: Montaje. Paso 43.

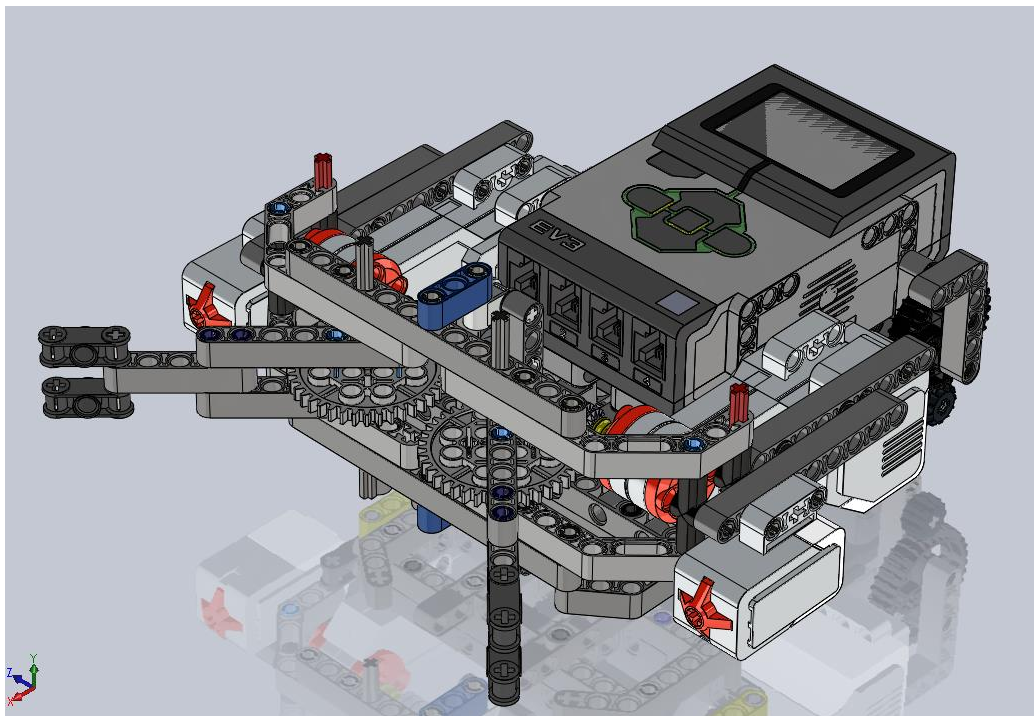


Ilustración 97: Montaje. Paso 44.

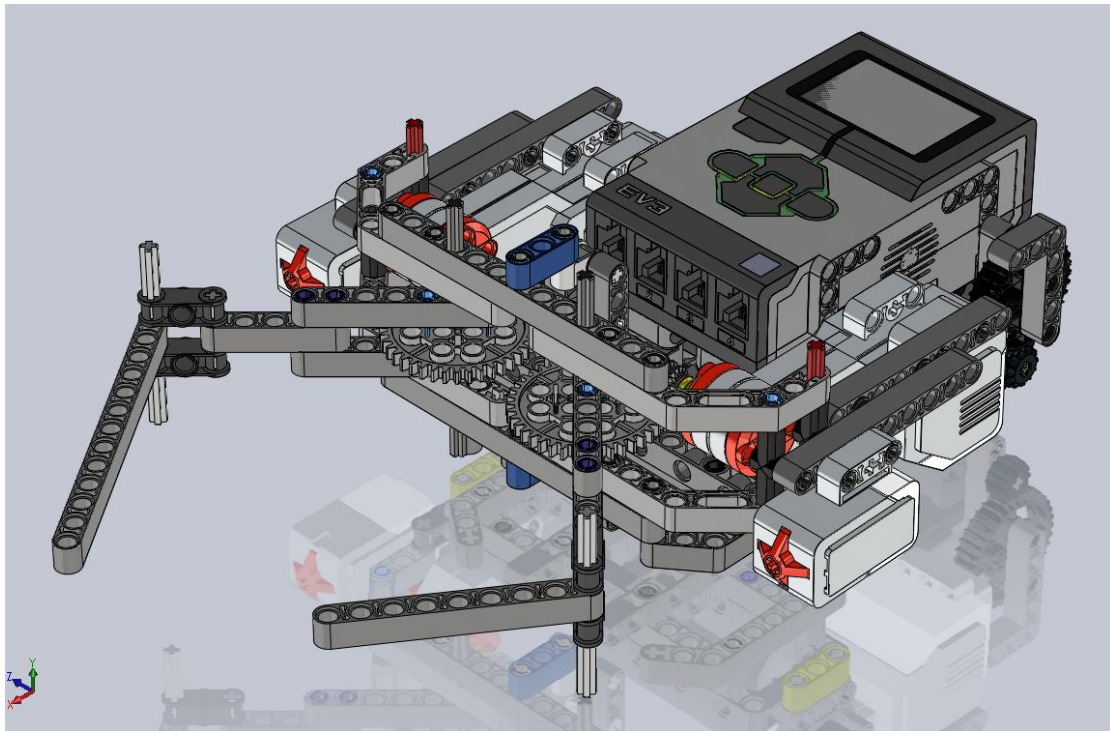


Ilustración 98: Montaje. Paso 45.

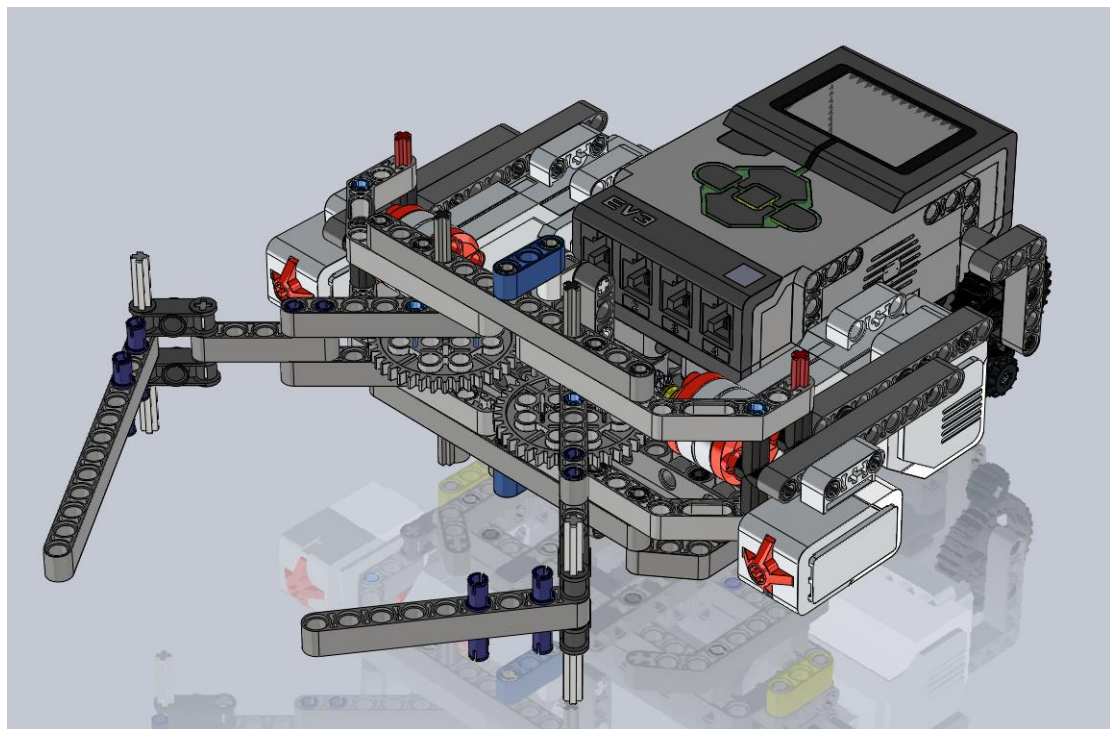


Ilustración 99: Montaje. Paso 46.

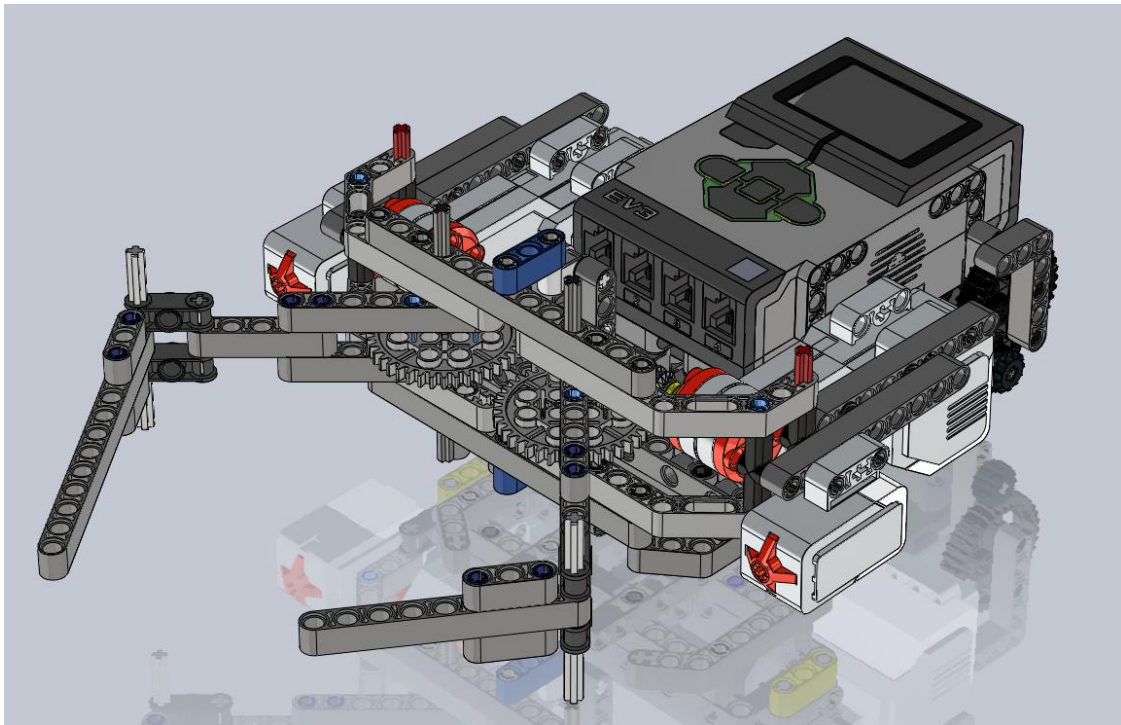


Ilustración 100: Montaje. Paso 47.

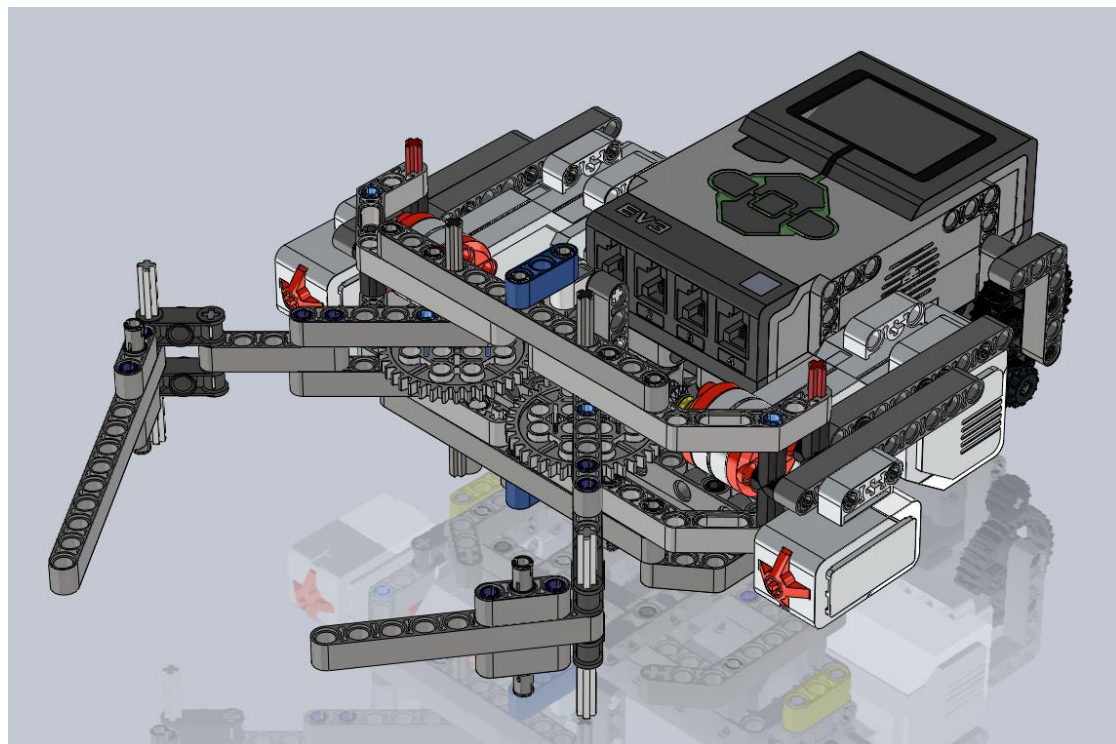


Ilustración 101: Montaje. Paso 48.

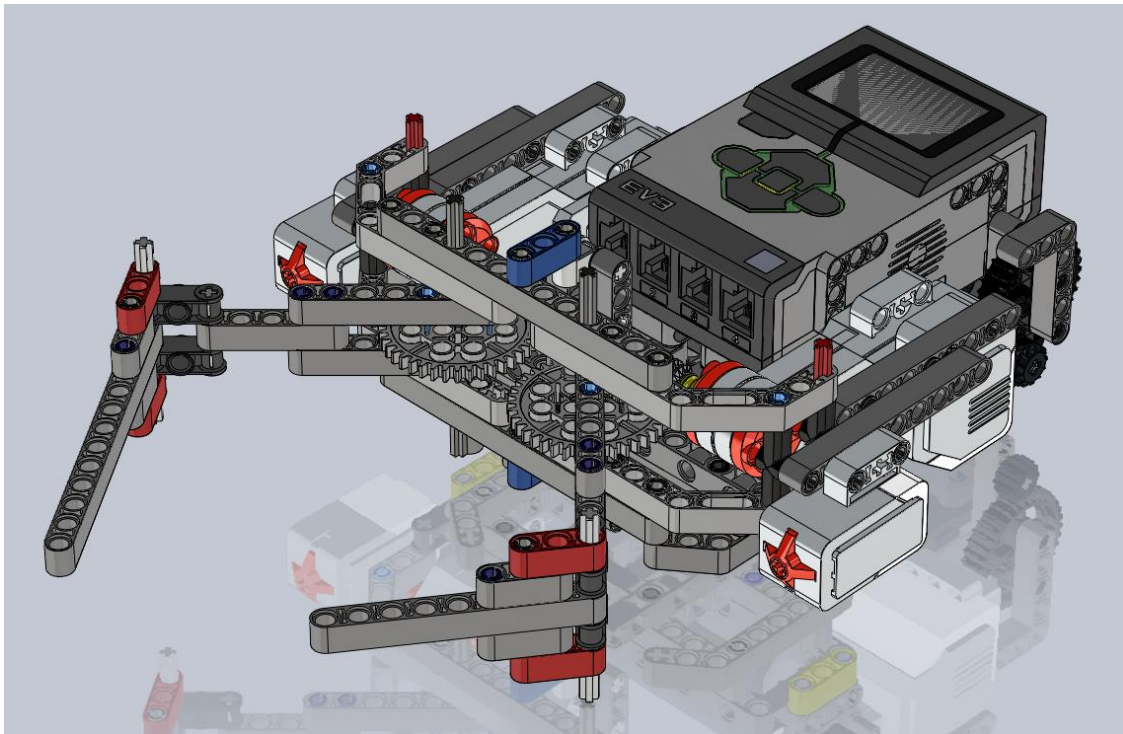


Ilustración 102: Montaje. Paso 49.

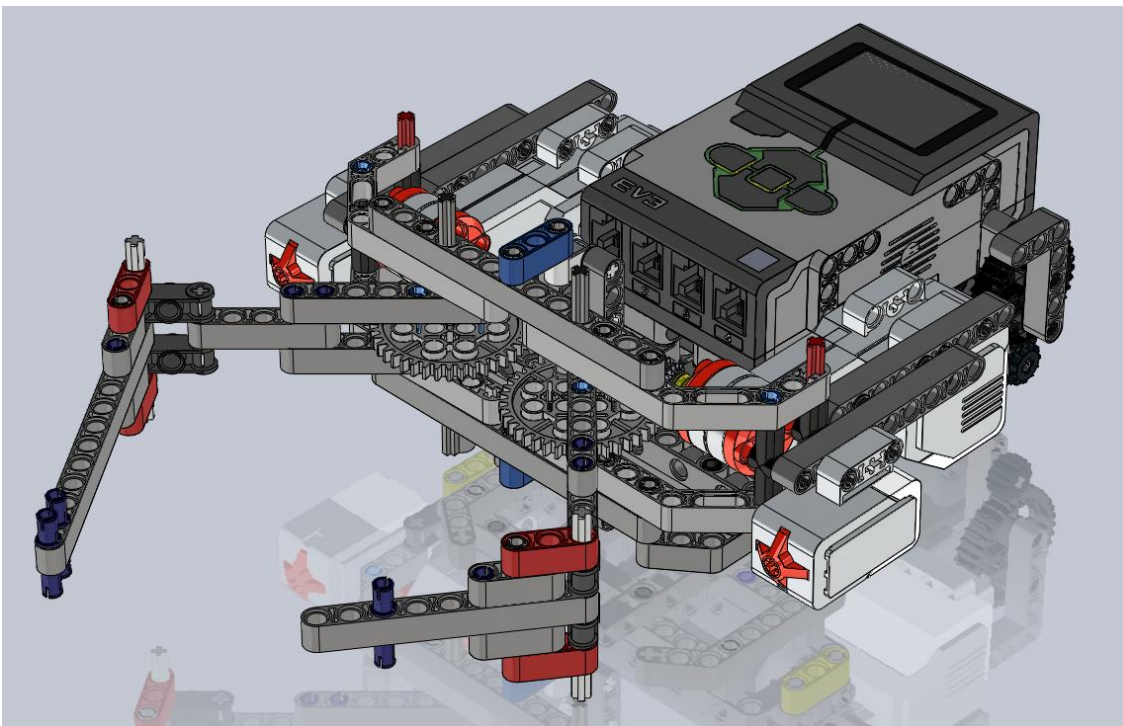


Ilustración 103: Montaje. Paso 50.

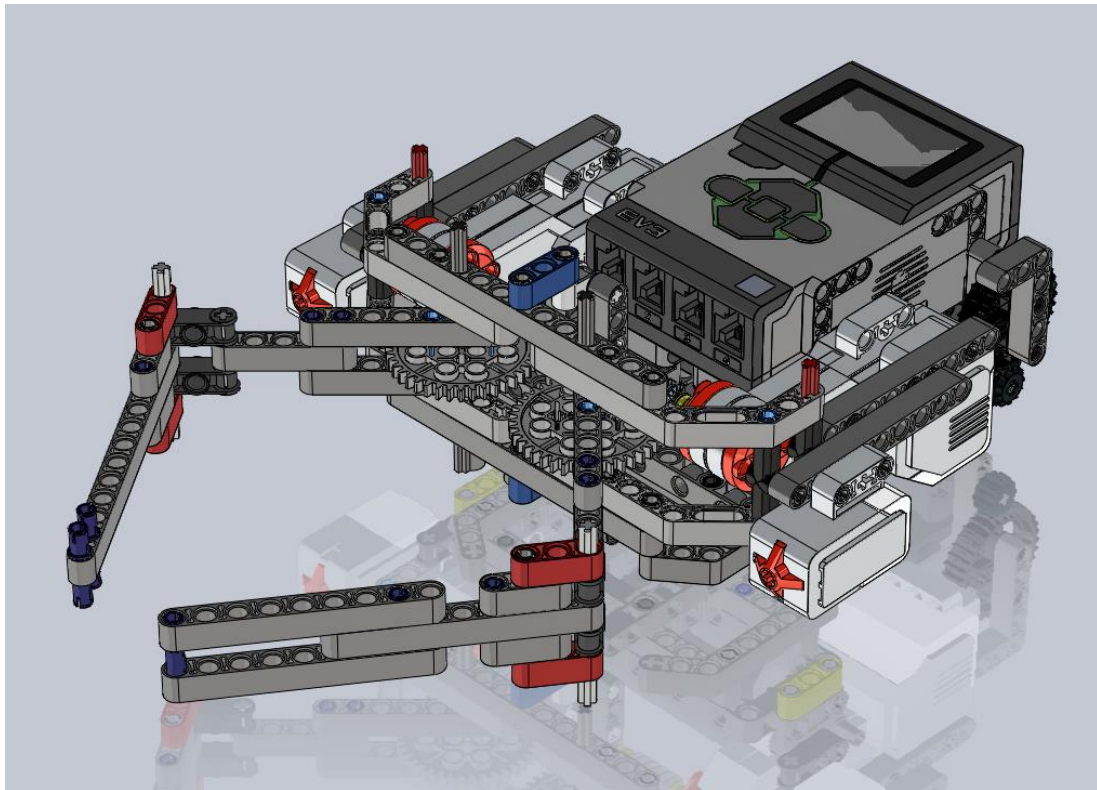


Ilustración 104: Montaje. Paso 51.

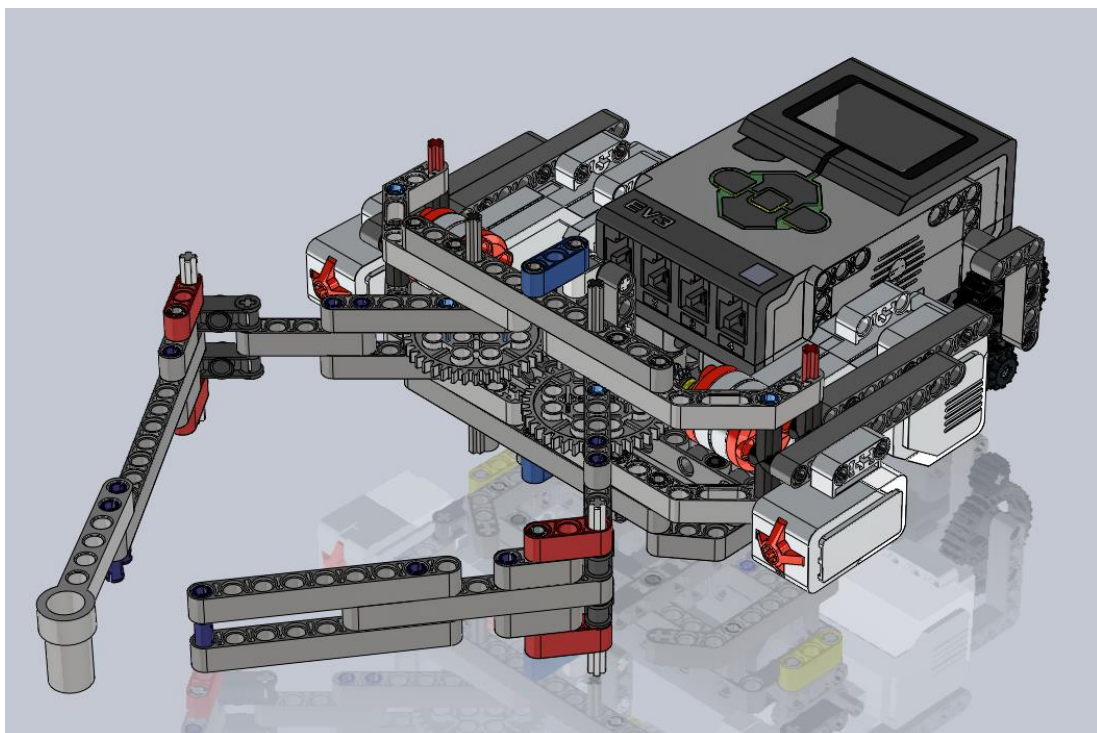


Ilustración 105: Montaje. Paso 52.

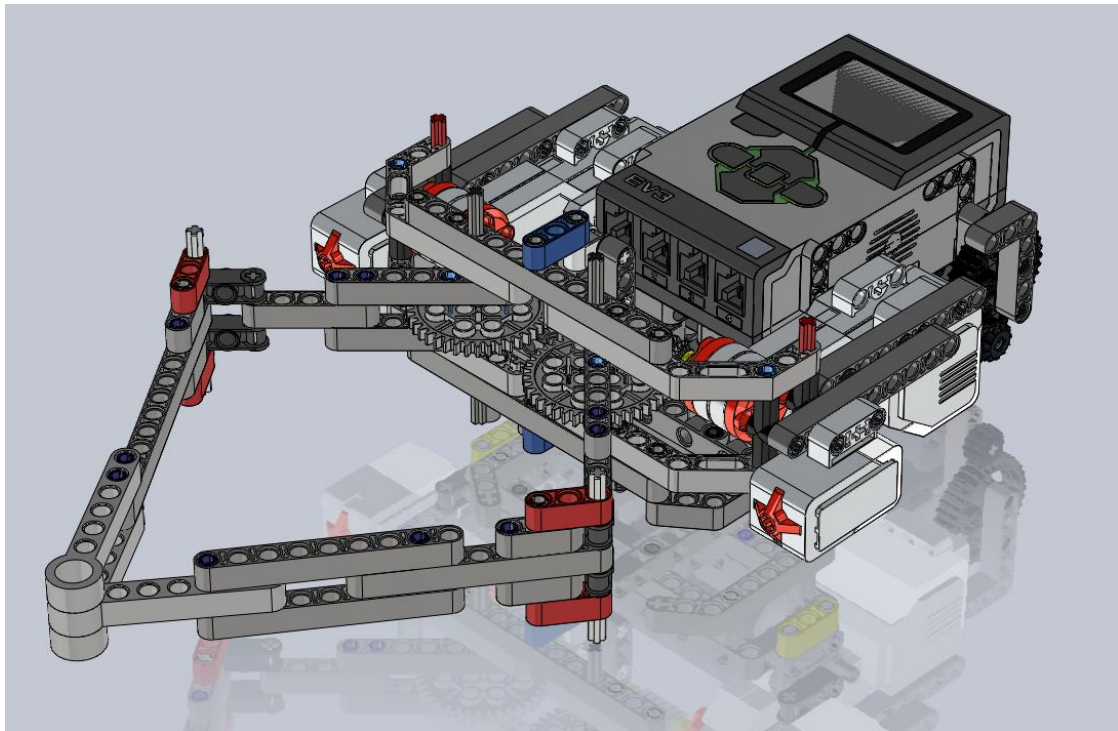


Ilustración 106: Montaje. Paso 53.

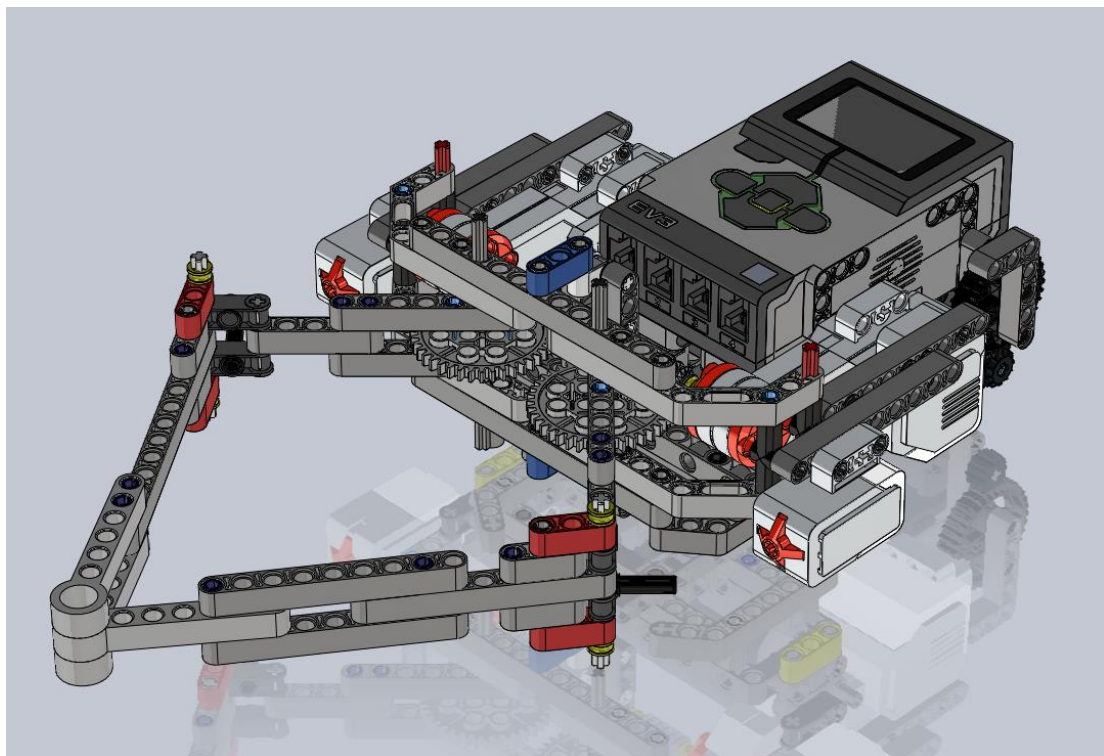


Ilustración 107: Montaje. Paso 54.





**UNIVERSIDAD
DE LA RIOJA**

Documento 4:

PLANOS

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

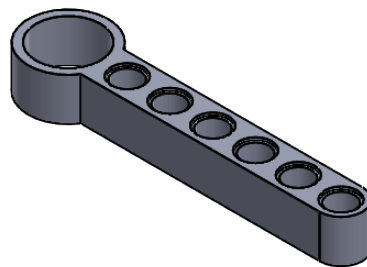
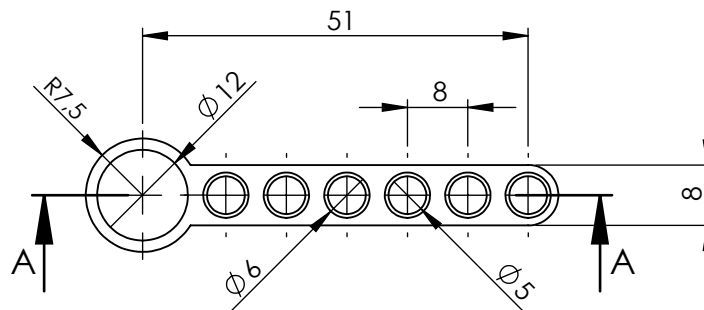
Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

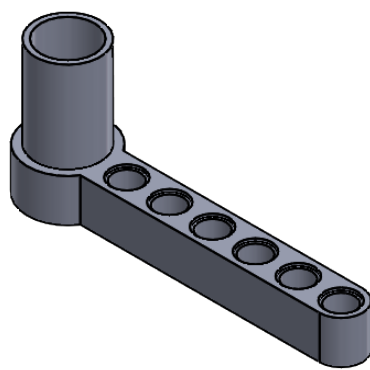
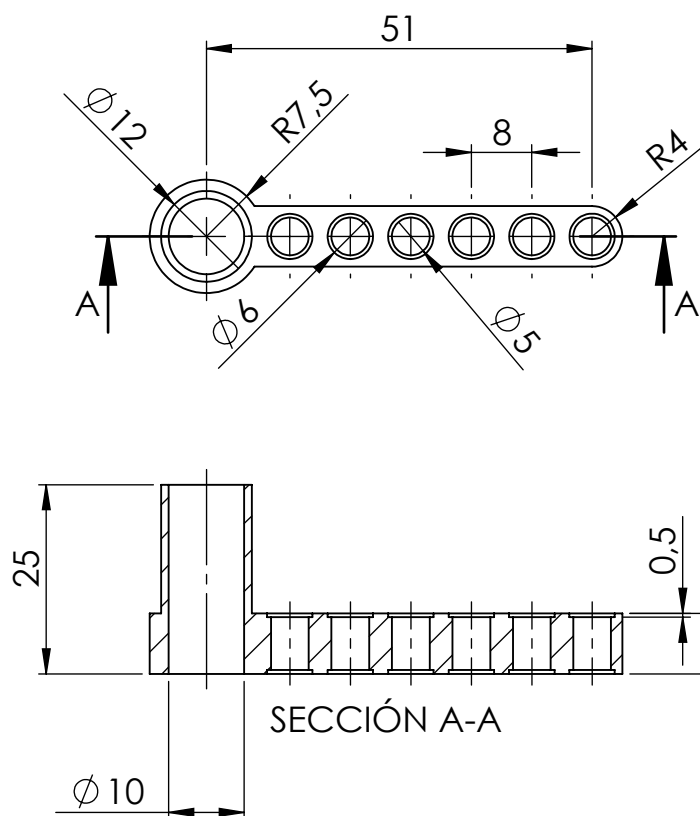
Tutor: Javier Rico Azagra



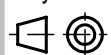
ÍNDICE DE PLANOS

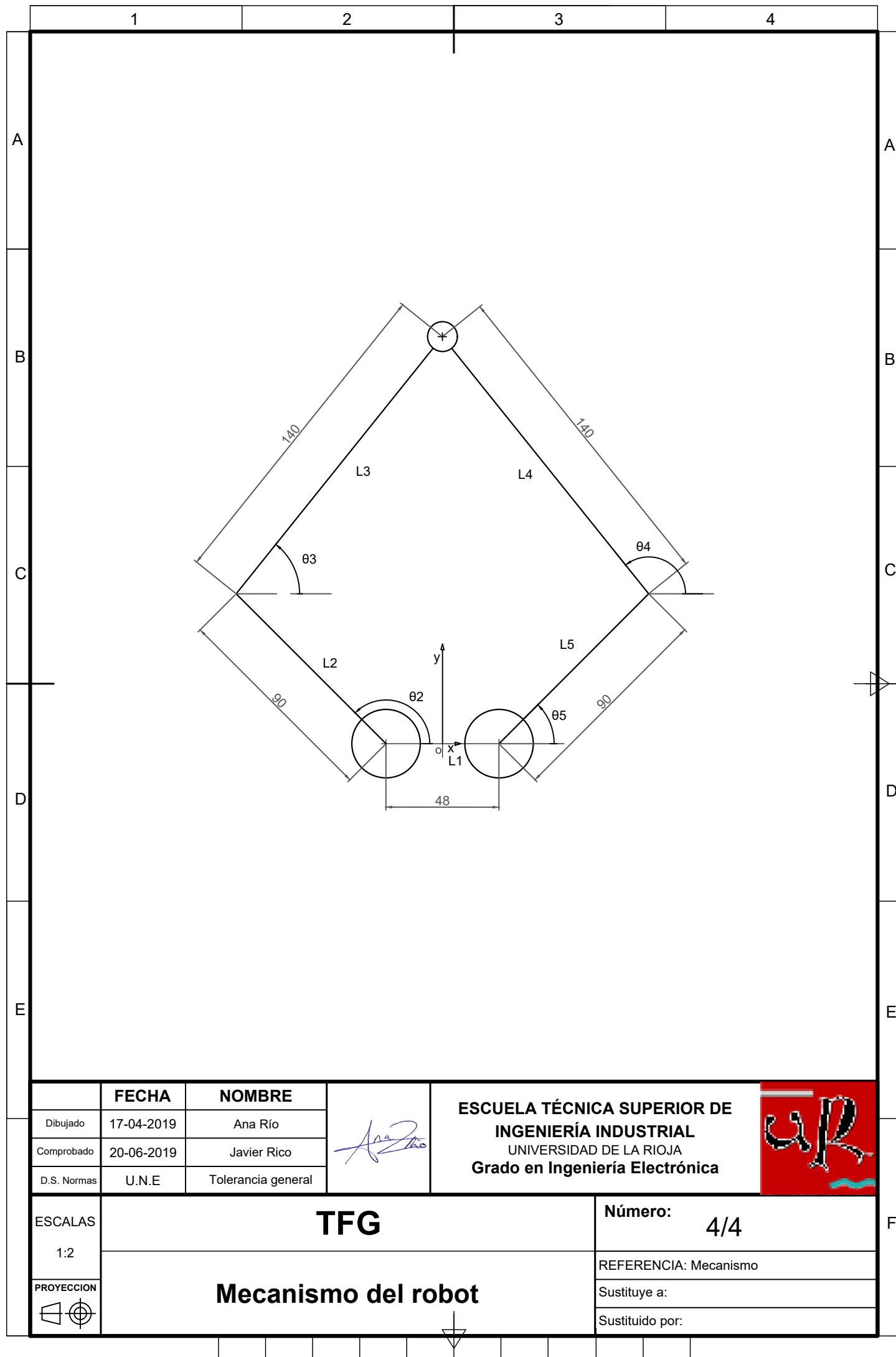
| | |
|--|-----|
| Plano Nº1: Pieza sujeción rotulador 1..... | 122 |
| Plano Nº2: Pieza sujeción rotulador 2..... | 123 |
| Plano Nº3: Dimensiones del sudoku..... | 124 |
| Plano Nº4: Mecanismo del robot..... | 125 |



| | | | | | |
|---|----------------------------|-------------|---|--|---|
| | Fecha | Nombre |  | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en Ingeniería Electrónica |  |
| Dibujado | 14-06-2019 | Ana Río | | | |
| Comprobado | 20-06-2019 | Javier Rico | | | |
| U.S.Norm.: | ISO | 2768-mK | | | |
| Escalas 1:1 | TFG | | | Número: 1/4 | |
| Proyeccion  | Pieza sujeción rotulador 1 | | | REFERENCIA: Pieza sujeción rotulador 1 | |
| | | | | Sustituye a: | |
| | | | | Sustituida por: | |



| | | | | | |
|---|----------------------------|-------------|---|--|---|
| | Fecha | Nombre |  | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL UNIVERSIDAD DE LA RIOJA Grado en Ingeniería Electrónica |  |
| Dibujado | 18-06-2019 | Ana Río | | | |
| Comprobado | 20-06-2019 | Javier Rico | | | |
| U.S.Norm.: | ISO | 2768-mK | | | |
| Escalas | TFG | | | Número: | 2/4 |
| 1:1 | | | | REFERENCIA: | Pieza rotulador 2 |
| Proyeccion | Pieza sujeción rotulador 2 | | | Sustituye a: | |
|  | | | | Sustituida por: | |







**UNIVERSIDAD
DE LA RIOJA**

Documento 5:

PLIEGO DE CONDICIONES

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

ÍNDICE DEL PLIEGO DE CONDICIONES

| | |
|---|-----|
| 1. INTRODUCCIÓN AL PLIEGO DE CONDICIONES..... | 129 |
| 2. PROPIEDAD INTELECTUAL | 129 |
| 3. CONDICIONES GENERALES | 129 |
| 4. CONDICIONES ADMINISTRATIVAS | 130 |
| 5. NORMATIVA Y REGLAMENTACIÓN | 130 |
| 6. CONDICIONES FACULTATIVAS..... | 131 |
| 6.1. Dirección | 131 |
| 6.2. Libro de órdenes | 131 |
| 6.3. Modificaciones | 131 |
| 7. CONDICIONES DE MATERIALES Y EQUIPOS..... | 132 |
| 7.1. Bloque de programación Lego EV3 | 132 |
| 7.2. Equipo PC..... | 132 |
| 8. CONDICIONES ECONÓMICAS | 133 |
| 8.1. Errores en el proyecto | 133 |
| 8.2. Liquidación..... | 133 |
| 9. DISPOSICIÓN FINAL | 134 |

1. INTRODUCCIÓN AL PLIEGO DE CONDICIONES

La autora de este proyecto ha cursado los estudios de Grado en Ingeniería Electrónica Industrial y Automática en la Universidad de La Rioja, cumpliendo con las directrices especificadas por la Escuela en la normativa trabajo fin de grado en vigor en el curso 2018/2019.

El objeto de este pliego de condiciones es recoger y establecer todas las disposiciones técnicas, administrativas y económicas, y las normativas que han de regir este proyecto.

El diseño de este proyecto y sus características han sido descritos en detalle a lo largo de la memoria y anexos.

En caso de no realizarse las condiciones tal como se presenta en este documento el proyectista no se responsabilizará de los posibles fallos y averías propias del funcionamiento, repercutiendo todo el peso del problema sobre terceras personas.

2. PROPIEDAD INTELECTUAL

Según el artículo 12 del reglamento PFC de la Escuela Técnica Superior de Ingeniería Industrial de La Rioja, aprobado por el Consejo de Gobierno del 15 de Abril de 2005, “Podrá ser consultada la copia de la biblioteca si el autor y el Director de Proyecto Fin de Carrera así lo autorizan por escrito”.

3. CONDICIONES GENERALES

Este proyecto se ajusta en su desarrollo a los reglamentos y disposiciones electrónicas vigentes. Atendiendo a esto, una vez se haya aprobado por el Ministerio de Industria, tendrá carácter de obligado cumplimiento.

Una vez realizado el proyecto, se podrán realizar diversas modificaciones siempre bajo la supervisión del ingeniero o proyectista.

En caso de efectuarse alguna modificación, el correspondiente proyecto modificado se considera como parte integrante del proyecto definitivo y como tal, sujeto a las condiciones y especificaciones citadas y aprobadas por el Ministerio.

La empresa adjudicataria suscribirá contrato ante notario donde se hará constar, a parte de los términos legales obligatorios, plazos de entrega y la conformidad con la sanción cuyo incumplimiento pueda acarrear.

4. CONDICIONES ADMINISTRATIVAS

El proyecto se compone de los siguientes documentos:

- Índice general: Indica la página en la que se da comienzo a todos y cada uno de los apartados del proyecto.
- Memoria: Donde se explican todos los procesos y requerimientos del proyecto. Así como la solución elegida.
- Anexos: Se recogen todos los documentos añadidos de interés para la correcta realización del proyecto.
- Planos: Se describen gráficamente las piezas no incluidas en el kit de Lego Mindstorms EV3, así como las dimensiones del *sudoku* y el robot.
- Pliego de condiciones: Se establecen las diferentes condiciones técnicas, económicas y administrativas.
- Mediciones: Se enumeran las cantidades necesarias de cada unidad de proyecto para la correcta realización del mismo.
- Presupuesto: Documento que recoge los precios unitarios y totales de todas las unidades de proyecto, ordenados por partidas.

5. NORMATIVA Y REGLAMENTACIÓN

La realización del proyecto se regirá por la Ordenanza General de Seguridad e Higiene en el Trabajo del 7 de abril de 1970 y posteriores actualizaciones. Así mismo, se regirá por el Reglamento Electrotécnico de Baja Tensión, en el que se tendrán en cuenta las siguientes normas:

- UNE 20-514-82: Reglas de seguridad para los aparatos electrónicos
- UNE 157001:2002: sobre criterios generales para la elaboración de proyectos, que establece las consideraciones generales que permitan precisar las características que deben satisfacer los proyectos de productos obras y edificios, instalaciones, servicios o software, para que sean conformes al fin que están destinados.

Este proyecto debido a sus características se encuentra recogido, dentro del reglamento electrónico de baja tensión:

“Se calificará como instalación eléctrica de baja tensión, todo conjunto de aparatos y circuitos asociados en previsión de un fin particular, producción, conversión, transformación, distribución o utilización de la energía eléctrica cuyas tensiones nominales sean iguales o inferiores a 1000 V para corriente alterna y 1500 V para corrientes continuas”.

6. CONDICIONES FACULTATIVAS

6.1. Dirección

La dirección del montaje estará realizada en su totalidad por el ingeniero o proyectista o por otra persona que delegue atendiendo a la capacidad de dicha persona para realizar este trabajo.

Una vez realizado el montaje, su utilización podrá ser realizada por cualquier persona con conocimientos suficientes sobre el proyecto, la tecnología implicada y su funcionamiento. En caso de avería o mal funcionamiento por incorrecta utilización, el proyectista queda exento de culpa.

6.2. Libro de órdenes

La utilización de las aplicaciones desarrolladas en el proyecto se deberá realizar atendiendo a los documentos según el siguiente orden de prioridad en el caso de que hubiera alguna contradicción:

1. Presupuesto
2. Pliego de condiciones
3. Memoria

6.3. Modificaciones

Si fuera necesario realizar alguna modificación en el presente proyecto, deberá comunicarse con anterioridad a la dirección del proyecto, quien deberá dar su autorización.

Si se realizan modificaciones sin ser comunicadas ni autorizadas por la dirección del proyecto, las consecuencias que dichos cambios ocasionen serán de total responsabilidad del instalador que las realice.

Las posibles modificaciones realizadas por el propietario del proyecto no serán tratadas de forma especial y, por tanto, en ningún caso quedan eximidas de la autorización de la dirección del proyecto.

En el caso de que surgiera alguna posible modificación aprobada por la dirección del proyecto el contratante será el encargado de abonar este importe íntegramente.

7. CONDICIONES DE MATERIALES Y EQUIPOS

En este apartado, se tratará de describir las características de todos los componentes de este proyecto.

7.1. Bloque de programación Lego EV3

- El Bloque EV3 requiere seis baterías AA/LR6. No combinar nunca diferentes tipos de baterías
- Se han de quitar las baterías del Bloque EV3 cuando no se esté utilizando.
- No se deben utilizar nunca baterías dañadas.
- Se debe utilizar el cargador de baterías correcto. transformador 10 VDC
- Nunca se ha de intentar recargar baterías que no sean recargables.
- Conectar actuadores en los puertos A, B, C o D.
- Conectar sensores en los puertos 1, 2, 3, o 4.
- Bajo ninguna circunstancia sumergir el bloque en ningún tipo de líquido.
- Si se realiza una conexión inalámbrica (WiFi o Bluetooth) entre el Bloque EV3 y el PC, se recomienda que la distancia entre ambos no supere los 10 metros para una correcta transmisión de datos.

7.2. Equipo PC

Para el desarrollo e implementación del proyecto se deberá disponer de un PC que será el encargado de recoger datos, analizarlos, procesarlos y enviarlos. Para ello deberá disponer al menos del siguiente software:

- Windows 10 – 32 o 64 bits
- Matlab – Simulink R2017b

8. CONDICIONES ECONÓMICAS

8.1. Errores en el proyecto

Ante cualquier fallo o error tanto de diseño como de funcionamiento se informará al proyectista para dar solución al problema.

En tal caso se recomienda el apagado inmediato del robot para evitar daños de los componentes, y no volver a reanudar la marcha hasta que no sea revisado por el proyectista.

8.2. Liquidación

Terminada la elaboración del proyecto se procederá a la liquidación, donde se incluyen los importes correspondientes a la elaboración del proyecto.

En el caso de que surgiera alguna posible modificación aprobada por la dirección del proyecto el contratante será el encargado de abonar este importe íntegramente.

Al suscribir el contrato el contratante deberá abonar al adjudicatario el 60% del total del presupuesto. Tras la entrega e instalación de la aeronave se deberá realizar un nuevo abono del 20% del total del presupuesto, que será condición necesaria para continuar con el proceso de diseño necesario para el control del UAV. En caso de no realizar el abono el proyectista queda exento de todos los deberes sus contractuales.

El 20% se quedará como garantía durante los 6 primeros meses a partir de la fecha de puesta en marcha del montaje final. Si transcurrido este periodo de tiempo no se han advertido evidencias de defecto, se abonará la cantidad pendiente de entrega, y a partir de ese momento, se consideran completamente concluidos los compromisos entre las dos partes, a excepción del periodo de garantía que cubre lo citado anteriormente.

9. DISPOSICIÓN FINAL

Las dos partes contratantes, dirección técnica y empresa, ratifican el contenido del siguiente pliego de condiciones, el cual tiene igual validez, a todos los efectos, que una estructura publica, prometiendo fiel cumplimiento.

Logroño a 11 de junio de 2019,



FIRMADO: Ana Río Rodríguez





**UNIVERSIDAD
DE LA RIOJA**

Documento 6:

MEDICIONES

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

ÍNDICE DE MEDICIONES

| | |
|-------------------------------|-----|
| 1. CUADRO DE MEDICIONES | 138 |
|-------------------------------|-----|

1. CUADRO DE MEDICIONES

| Capítulo | Código | Unidad de proyecto | Unidades |
|----------------------|--------|---|----------|
| Recursos humanos | C0.1 | Ingeniero Electrónico (programación) | 90 h |
| | C0.2 | Técnico de montaje | 5 h |
| | C0.3 | Diseñador de planos y modelo 3D | 40 h |
| | C0.4 | Proyectista | 50 h |
| | | | |
| Licencias | C1.1 | Amortización Licencia de Windows 10-64bits | 180 h |
| | C1.2 | Amortización Licencia Matlab R2017a | 90 h |
| | C1.3 | Amortización Licencia AutoCAD | 10 h |
| | C1.4 | Amortización Licencia Solidworks | 30 h |
| | | | |
| Equipos y materiales | C2.1 | Ordenador portátil | 1 ud |
| | C2.2 | LEGO Mindstorms transformador 10 VDC | 1 ud |
| | C2.3 | Set principal LEGO Mindstorms Education EV3 | 1 ud |
| | C2.4 | Set expansión LEGO Mindstorms Education EV3 | 1 ud |
| | C2.5 | Impresión de pieza en 3D | 14,96 g |
| | C2.6 | Rotulador Staedtler permanente F | 1 ud |
| | C2.7 | Conector de WiFi tipo USB Netgear N150 | 1 ud |
| | C2.8 | Impresión plantilla del <i>sudoku</i> | 10 hojas |





**UNIVERSIDAD
DE LA RIOJA**

Documento 7:
PRESUPUESTO

**DESARROLLO DE SISTEMA AUTÓNOMO BASADO EN
LA PLATAFORMA LEGO EV3 PARA LA RESOLUCIÓN
DE JUEGOS MATEMÁTICOS**

Grado en Ingeniería Electrónica, Industrial y Automática

Escuela Técnica Superior de Ingeniería Industrial

Curso académico: 2018/2019

Alumna: Ana Río Rodríguez

Tutor: Javier Rico Azagra

ÍNDICE DEL PRESUPUESTO

| | |
|-------------------------------------|-----|
| 1. CUADRO DE PRECIOS UNITARIOS..... | 142 |
| 2. PRESUPUESTOS PARCIALES | 143 |
| 2.1. Recursos humanos | 143 |
| 2.2. Licencias..... | 143 |
| 2.3. Equipos y materiales..... | 144 |
| 3. PRESUPUESTO TOTAL | 145 |

1. CUADRO DE PRECIOS UNITARIOS

| Capítulo | Código | Unidad de proyecto | Precio unitario |
|----------------------|--------|---|-----------------|
| Recursos humanos | C0.1 | Ingeniero Electrónico (programación) | 25,00 €/h |
| | C0.2 | Técnico de montaje | 10,00 €/h |
| | C0.3 | Diseñador de planos y modelo 3D | 20,00 €/h |
| | C0.4 | Proyectista | 15,00 €/h |
| | | | |
| Licencias | C1.1 | Amortización Licencia de Windows 10-64bits | 0,01 €/h |
| | C1.2 | Amortización Licencia Matlab R2017a | 0,09 €/h |
| | C1.3 | Amortización Licencia AutoCAD | 0,24 €/h |
| | C1.4 | Amortización Licencia Solidworks | 0,28 €/h |
| | | | |
| Equipos y materiales | C2.1 | Ordenador portátil | 830,00 €/ud |
| | C2.2 | LEGO Mindstorms transformador 10 VDC | 30,88 €/ud |
| | C2.3 | Set principal LEGO Mindstorms Education EV3 | 399,00 €/ud |
| | C2.4 | Set expansión LEGO Mindstorms Education EV3 | 111,50 €/ud |
| | C2.5 | Impresión de pieza en 3D | 0,20 €/g |
| | C2.6 | Rotulador Staedtler permanente F | 2,29€/ud |
| | C2.7 | Conector de WiFi tipo USB Netgear N150 | 20,18 €/ud |
| | C2.8 | Impresión plantilla del <i>sudoku</i> | 0,05 €/hoja |

2. PRESUPUESTOS PARCIALES

2.1. Recursos humanos

| Unidad de proyecto | Código | Precio unitario | Mediciones | Precio parcial |
|--------------------------------------|--------|-----------------|------------|----------------|
| Ingeniero Electrónico (programación) | C0.1 | 25,00 €/h | 90 h | 2.250,00 € |
| Técnico de montaje | C0.2 | 18,00 €/h | 5 h | 90,00 € |
| Diseñador de planos y modelo 3D | C0.3 | 20,00 €/h | 40 h | 800,00 € |
| Proyectista | C0.4 | 22,00 €/h | 50 h | 1.100,00 € |
| Suma total | | | | 4.240,00 € |

La cantidad asciende a un precio de CUATRO MIL DOSCIENTOS CUARENTA EUROS.

2.2. Licencias

| Unidad de proyecto | Código | Precios unitarios | Mediciones | Precio parcial |
|---|--------|-------------------|------------|----------------|
| Amortización Licencia Windows 10-64bits | C1.1 | 0,01 €/h | 180 h | 1,80 € |
| Amortización Licencia Matlab R2017a | C1.2 | 0,09 €/h | 90 h | 8,10 € |
| Amortización Licencia AutoCAD | C1.3 | 0,24 €/h | 10 h | 2,40 € |
| Amortización Licencia Solidworks | C1.4 | 0,28 €/h | 30 h | 8,40 € |
| Suma total | | | | 20,70 € |

La cantidad asciende a un precio de VEITE EUROS CON SETENTA CÉNTIMOS.

2.3. Equipos y materiales

| Unidad de proyecto | Código | Precio unitario | Mediciones | Precio parcial |
|---|--------|-----------------|------------|----------------|
| Ordenador portátil | C2.1 | 229,00 €/ud | 1 ud | 830,00 € |
| LEGO Mindstorms transformador 10 VDC | C2.2 | 30,88 €/ud | 1 ud | 30,88 € |
| Set principal LEGO Mindstorms Education EV3 | C2.3 | 399,00 €/ud | 1 ud | 399,00 € |
| Set expansión LEGO Mindstorms Education EV3 | C2.4 | 111,50 €/ud | 1 ud | 111,50 € |
| Impresión de pieza en 3D | C2.5 | 0,20 €/g | 14,96 g | 2,99 € |
| Rotulador Staedtler permanente F | C2.6 | 2,29 €/ud | 1 ud | 2,29 € |
| Conector de WiFi tipo USB Netgear N150 | C2.7 | 20,18 €/ud | 1 ud | 20,18 € |
| Impresión plantilla del <i>sudoku</i> | C2.8 | 0,05 €/hoja | 10 hojas | 0,50 € |
| Suma total | | | | 1.397,34 € |

La cantidad asciende a un precio de MIL TRESCIENTOS NOVENTA Y SIETE EUROS CON TREINTA Y CUATRO CÉNTIMOS

3. PRESUPUESTO TOTAL

| Capítulo | Precio |
|---------------------------|-------------------|
| Recursos humanos | 4.240,00 € |
| Licencias | 20,70 € |
| Equipos y materiales | 1.397,34 € |
| | |
| TOTAL Capítulos | 5.658,04 € |
| | |
| Beneficio industrial (%6) | 339,48 € |
| | |
| IVA (21%) | 1.259,48 € |
| | |
| COSTE TOTAL | 7.257 € |

El presupuesto total, asciende a la cantidad de: SIETE MIL DOSCIENTOS CINCUENTA Y SIETE EUROS.

Este presupuesto se ha estimado para una sola unidad de venta sin tener en cuenta descuentos por volúmenes de compra. Además, dichos precios son los establecidos a día de hoy 11 de Junio de 2019 pudiendo variar los mimos con el paso del tiempo.

Logroño a 11 de Junio de 2019,



FIRMADO: Ana Río Rodríguez





**UNIVERSIDAD
DE LA RIOJA**

TFG

Ana Río Rodríguez